Digital Displacement Hydrostatic Transmission Systems

Niall James Caldwell

A Thesis Submitted for the Degree of Doctor of Philosophy

The University of Edinburgh

2007

To my father.

Declaration

I certify that the work presented in this thesis, except where explicitly credited to others, is of my own commission both in substance and composition, and that it has not been submitted for any other degree or professional qualification.

N.J. Caldwell

Abstract

Digital Displacement pumps and motors are a new type of hydraulic machine, in which fluid commutation and displacement control are achieved by solenoid actuated valves under the command of a microprocessor, rather than mechanical means. The thesis is that radial piston machines, built according to this principle, offer energy efficiency and control advantages over variable stroke axial piston pumps, when applied to hydrostatic vehicle transmissions.

Experimental results on the efficiency of prototypes are analysed and compared to published results from swashplate machines, showing an improvement in energy efficiency. Loss models are proposed and compared with experiment.

A Digital Displacement motor suitable for propelling a vehicle is described and the design and development of the mechanics, electro-magnetics and embedded software are described. Experimental results are also presented, illustrating the performance of a demonstrator vehicle driven by the motor, in particular demonstrating the closed-loop regulation of vehicle speed using motor displacement control.

A demonstrator vehicle is described which features a hydrostatic transmission using both a Digital Displacement pump and an axial piston motor. Experimental results of pump performance are presented with specific focus on vehicle propel. A control technique is described which increases the sensitivity of the pump at low speeds. Results are presented of tests on the prototype transmission system, focussing on the time-domain system dynamics. A computer simulation model of the vehicle is presented and results compared to experiment.

Acknowledgements

Designing and making DD machines requires a mix of abilities, from mechanical design and practical prototyping, to electronics and embedded software. I am grateful to Stephen Salter and Jamie Taylor for the experience of working as part of the Wave Power Group at the University of Edinburgh, between January 1996 and January 1999, which equipped me with many of the skills needed to tackle the work presented here.

I am indebted to the whole Artemis Intelligent Power Ltd. team, without which this work would not have been possible. Win Rampen provided invaluable supervision and ideas throughout. Jon Almond designed the electronic hardware and the low-level embedded software for the DDP used in the golf buggy demonstrator vehicle, and the electronic hardware for the propel DDPM. Carn Gibson designed the mechanical interface between the DDP and the golf buggy engine. Pierre Joly and Jack Lavender designed and commissioned the SD1B machine. Gordon Voller designed and commissioned the C2 machine. Fergus McIntyre developed the solenoid valves for the C2 machine. For the SD1B and C2 machines, Uwe Stein led the mechanical design, Michael Fielding designed the electronics and embedded software, and Bill Tullis made many of the prototype parts.

The support of Sauer-Danfoss was crucial to the work presented here. I greatly appreciate the personal commitment of Ken Lai, Onno Kuttler and Luke Wadesley, and their families, in moving to Edinburgh to work with AIP on commercialisation of DD technology. Luke Wadesley organised the test program of the SD1B at the Sauer-Danfoss test facility. Onno Kuttler has been particularly supportive of this thesis and continues to lead the commercialisation of DD technology for Sauer-Danfoss. The guidance of Leif Tandrup and Jeff Herrin has been important to the ongoing effort to bring the technology to the mobile hydraulic market.

Notation

196	Intel 80196 microcontroller
A	Area
ADC	Analogue to digital converter
AIP	Artemis Intelligent Power Ltd.
AT	The ampere-turns of a coil
axgap	Of a solenoid, the axial gap between armature and the flux concetrator
B	Magnetic flux density
BDC	Bottom dead centre; the instant of maximum volume
C167	Infineon C167 microcontroller
C2	The code name of the single bank DDP, described in Section 2.8
C_f	Friction loss coefficient
$C_{pressure}$	Coefficient describing behaviour of leakage with respect to pressure
C_r	Of a hydrostatic bearing, the clamping ratio
C_s	Slip loss coefficient
C_{speed}	Coefficient describing behaviour of leakage with respect to speed
C_v	Viscous loss coefficient
coiler	Of a solenoid, the external radius of the coil
coilir	Of a solenoid, the internal radius of the coil
current	Of a solenoid, the current in the coil
CN	Compliance number; the number of full strokes of a DDP needed to raise the system pressure from zero to maximum
CNC	Computer numerically controlled
D	Pump displacement
DAC	Digital to analogue converter
DAQ	Data acquisition system

DD	Digital Displacement[1]
DDP	Digital Displacement Pump
DDPM	Digital Displacement Pump/Motor
DDPM-P	A DDPM capable of propelling a vehicle
e.m.f.	Electromotive force
F	Force
F_d	Fraction of maximum displacement
FE	Finite element
FEA	Finite element analysis
FET	Field effect transistor
h	Clearance
H	Magnetic field intensity
HG	Of a DDP controller, the event of starting the pulse to the HPV
HP	High pressure
$hp_{(0,1,2,3)}$	Coefficients of P_{hploss}
HPV	High pressure valve; the delivery valve
HS	Of a DDP controller, the event of ending the pulse to the HPV
i	Current
$id_{(0,1,2,3)}$	Coefficients of $P_{idleloss}$
k_{damp}	Of a solenoid, the mechanical damping constant of the armature
k_{dead}	Ratio between V_{dead} and V_{geom} of a pump
$k_{leakscale}$	Scale factor applied to model pump losses relative to Q_{hp0}
k_{speed}	Constant describing the behavior of pump leakage with respect to speed
k_{spring}	Of a solenoid, the spring rate
$lp_{(0,1,2,3)}$	Coefficients of P_{lploss}
L	Magnetic inductance

[1] 'Digital Displacement' is a trademark of Artemis Intelligent Power Ltd.

L_{inc}	Incremental magnetic inductance
LED	Light-emitting diode
LG	Of a DDP controller, the event of starting the pulse to the LPV
LP	Of a valve or pump, low pressure; of a mathematical model, lumped-parameter
LPV	Low pressure valve; the intake valve
LS	Of a hydraulic pump or system, load sensing; of a DDP controller, the event of ending the pulse to the LPV
LSB	Least significant bit
M	Mass
MSB	Most significant bit
n	Of a pump, rotational speed (revolutions per second)
N	Number of turns of a coil
NC	Of a valve, normally closed
NO	Of a valve, normally open
p	Pressure
p_{max}	Maximum pressure for the DDP loss model
p_{zero}	The pressure at which the leakage gap would close completely
P	Power
$P_{fulldisp.loss}$	The polynomial function of ω and p describing total losses at full displacement, at a given output pressure
$P_{idleloss}$	The polynomial function of ω describing total losses at zero displacement
P_{hploss}	The polynomial function of ω describing total losses at full displacement, maximum output pressure
P_{in}	The input mechanical power of the DDP
P_{lploss}	The polynomial function of ω describing total losses at full displacement, zero output pressure
P_{loss}	Total loss power

$P_{idleloss}$	The polynomial function of ω describing total loss power in the idle condition (F_d=0)
$P_{lossmech}$	Mechanical loss power
$P_{lossvol}$	Volumetric loss power
$P_{totalloss}$	The polynomial function of ω, p and F_d describing total losses at full displacement, at a given output pressure and displacement fraction
PC	Personal computer
PCB	Printed circuit board
PI	Proportional-integral controller
PID	Proportional-integral-derivative controller
poler	Of a solenoid, the external radius of the central pole.
poleir	Of a solenoid, the internal radius of the poppet guide bore in the central pole.
ppr	Pulses per revolution
preload	Of a solenoid, the spring compression when the armature is fully returned
PS	Part stroke
PWM	Of a solenoid, pulse width modulation; high-frequency excitation with voltage pulses
r_i	Of a hydrostatic bearing, the radius of the inside of the pad area
r_o	Of a hydrostatic bearing, the radius of the piston area
r_p	Of a hydrostatic bearing, the radius of outside of the pad area
R	Electrical resistance
R_c	Coil electrical resistance
R_f	Electrical resistance of the 'flywheel resistor' in a solenoid drive circuit
rgap	Of a solenoid, the radial gap between armature and the radial flux ring
ringir	Of a solenoid, the internal radius of the radial flux ring
RS232	A standard serial digital communications interface
Q	Flow rate
Q_{comp}	Flow rate missing from the DDP outlet due to compressibility
Q_{hp0}	The leakage flow equivalent to hp_0 divided by p_{max}

Q_{leak}	Flow rate missing from the DDP outlet due to leakage
Q_{loss}	Loss flow rate
Q_{nom}	The nominal expected flow from the DDP before the effects of leakage and compressibility
Q_{th}	Extrapolated flow rate at zero pressure differential of a pump
Q_{miss}	Missing flow rate - deviation of actual flow rate from Q_{th}
SC	Of a valve, solenoid-closed
SD1B	The code name of the three bank DDP, described in Section 2.5
SO	Of a valve, solenoid-opened
t	Time
t_{dA}	Of a solenoid valve, delay time for actuation
t_{dR}	Of a solenoid valve, delay time for release
t_{tA}	Of a solenoid valve, transit time for actuation
t_{tR}	Of a solenoid valve, transit time for release
t_p	Of a DDP controller, duration of the exciting voltage pulse

(subcripts $_{-HP}$ and $_{-LP}$ of the above refer to the HPV and LPV solenoids respectively)

T	Torque
T_E	Constant torque loss
T_{in}	Input torque of the DDP
T_{loss}	Loss torque
T_{theory}	Theoretical torque of a pump
taperh	Of a solenoid, the height of the flux concentrator
taperw	Of a solenoid, the width of the flux concentrator
thk	Of a solenoid, the axial thickness of the armature
TDC	Top dead centre; the instant of minimum volume
V	Volume; voltage
V_c	Of a solenoid, the coil voltage

V_{dead}	The total dead volume - the pressurised volume inside all the working chambers of the DDP which is not swept volume
V_{eff}	Effective output displacement of the DDP
V_{geom}	Geometric displacement of a pump
V_r	Ratio between dead volume in the chamber and the swept volume
V_i	Derived capacity (displacement) of a pump
V_{comp}	Displacement missing due to compressibility
V_{leak}	Displacement missing due to leakage
V_{miss}	Difference between actual displacement and V_i
V_{motion}	Of a solenoid coil, the motional electromotive force
V_{nom}	The nominal displacement of the DDP
x	position; fraction of maximum swash angle
x_{norm}	function of p_{zero} and p describing the nomalised leakage gap
$x_{squeeze}$	Of a solenoid, the position at which the squeeze-film effect starts
α	Of a hydrostatic bearing, the swashplate angle
β	Fluid bulk modulus
η	Efficiency
$\eta_{overall}$	Overall efficiency
η_{vol}	Volumetric efficiency
η_{mech}	Mechanical efficiency
λ	Magnetic flux linkage of a coil
μ	Dynamic fluid viscosity
μ_0	Magnetic permeability of free space
ρ	Fluid density
τ	Time constant
ω	Rotational speed (rad/s)
ω_{max}	Maximum pump rotational speed (rad/s)

Table of Contents

1 Introduction 1
1.1 Review of DD fundamentals 1
1.2 Literature on DD technology 3
1.3 Thesis 8
1.4 Chapter summary 9
1.5 Chronology 10

2 Efficiency of the DDP 11
2.1 Introduction 11
2.2 Literature review of loss models of hydrostatic machines 12
2.3 Background to DDP loss models 16
2.4 Analysis of DDP test results 17
2.5 The SD1B test machine 22
2.6 The proposed DDP loss model 52
2.7 Modelling SD1B results 61
2.8 The C2 test machine 69
2.9 Modelling C2 results 73
2.10 Conclusions 85

3 Development of valves for the DDPM 87
3.1 The DDPM concept 87
3.2 Early DDPM development 101
3.3 Development of the LPV for the propel DDPM 120
3.4 Magnetostatic finite element analysis and parametric design improvement of the HPV 126
3.5 Transient analysis of the HPV 153
3.6 Conclusions 172

4 The DDPM propel demonstrator 173
4.1 Introduction 173
4.2 Secondary control of hydraulic motors 173
4.3 DDPM vehicle transmission concepts 175
4.4 Overall mechanical design 177
4.5 Hydrostatic bearings 180
4.6 Electronic system 194
4.7 Motor software development 198
4.8 Vehicle demonstrator 218
4.9 Vehicle testing and development 220
4.10 Results 223
4.11 Conclusions 227

5 The DDP propel demonstrator vehicle 229
5.1 Introduction 229
5.2 Description of system 231
5.3 Interface and control design 241
5.4 Experimental results 265
5.5 Conclusions 288

6 Simulation of the DDP propel system 289
6.1 Introduction 289
6.2 Literature review 292
6.3 Implementation of the hydraulic system model 295
6.4 Identification of important system parameters 303
6.5 The DDP model 310

6.6 Results with an ideal pump flow source 325
6.7 Results with a detailed DDP model 330
6.8 Conclusions 346

7 Conclusions 349
7.1 Recommendations for further work 350

8 References 351

9 Appendices 361
9.1 Measuring flow at low pressure for efficiency tests 361
9.2 Special considerations when testing DDP efficiency 362
9.3 Applying the DDP loss model to backward-facing simulation studies 370
9.4 Development of the LPV poppet attachment 374
9.5 Mathcad calculations for capillary impedance design 376
9.6 ANSYS input deck for the HPV analysis 377
9.7 Wheel motor embedded software 381
9.8 DDP propel demonstrator - system controller software 412
9.9 Effect of a broken cylinder on DDP flow quality 433

List of Figures

Page no.

Figure 1-1: Radial section and exploded view of a radial-piston DDPM 2
Figure 1-2: Section of a radial-piston DDP, normal to the axis 3
Figure 1-3: The single cylinder DDPM by Rampen et al., 1994. (courtesy Win Rampen) 6
Figure 2-1: Uncertainty in the split of "missing flow" between the effects of leakage and compressibility 20
Figure 2-2: Isometric view of the SD1B overall design 23
Figure 2-3: Section view showing three-bank design of the SD1B 23
Figure 2-4: Flow lines in a single SD1B working chamber 24
Figure 2-5: Table of important design parameters of the SD1B 25
Figure 2-6: Pierre Joly calibrating the SD1B controller in the AIP lab 26
Figure 2-7: The SD1B being commissioned as an electronic load-sensing pump at the AIP lab in November 2003. From left to right: Uwe Stein, the author, Jack Lavender 26
Figure 2-8: Tests on an aerial lift fitted with the SD1B working as an electronic load-sensing open-circuit pump 27
Figure 2-9: Test rig for the SD1B tests 28
Figure 2-10: SD1B test plan 30
Figure 2-11: SD1B; Results from test 121 (filtered signals from transducers) 31
Figure 2-12: SD1B; basic analysis from test 121 32
Figure 2-13: SD1B; Flow vs. speed, lines of pressure (bar) 33
Figure 2-14: SD1B: V_{eff} vs. speed; lines of pressure 34
Figure 2-15: SD1B: V_{eff} vs. pressure; lines of speed 35
Figure 2-16: SD1B; Total losses as a function of speed (rad/s); lines of pressure 36
Figure 2-17: SD1B; Total losses as a function of pressure; lines of speed (rpm) 37
Figure 2-18: SD1B; Calculated instantaneous idle losses per cylinder due to LPV breathing at 2000rpm. 38
Figure 2-19: SD1B; idle torque loss analysis 39
Figure 2-20: Calculating the pressurised volume of the SD1B with Solidworks 2003 40
Figure 2-21: SD1B; Derived leakage flow (l/min) as a function of pressure (bar); lines of speed (rpm) 41
Figure 2-22: SD1B; Derived leakage flow as a function of speed; lines of pressure 42
Figure 2-23: SD1B; Derived torque loss as a function of speed (rpm); lines of pressure (bar) 43
Figure 2-24: SD1B; Derived torque loss as a function of pressure; lines of speed (rpm) 44
Figure 2-25: SD1B; Overall efficiency as a function of speed; lines of pressure (bar) 45
Figure 2-26: SD1B; Volumetric and mechanical efficiency by revised definitions as a function of speed; lines of pressure 46
Figure 2-27: SD1B; Variation of P_{loss} as a function of F_d; lines of rpm 47
Figure 2-28: SD1B; variation of Q and P_{loss} as a function of F_d; 1800rpm, 5 bar load 48
Figure 2-29: SD1B; variation of P_{loss} as a function of F_d at a range of pressures; 2000rpm 49
Figure 2-30: SD1B; variation of input torque as a function of F_d at a range of pressures (bar); 2000rpm 50
Figure 2-31: Extrusion of seals in the SD1B, causing parasitic leakage 51
Figure 2-32: Forward-facing (left) and backward-facing (right) pump loss models 53
Figure 2-33: The three polynomial curves which describe losses 54
Figure 2-34: Linear interpolation of losses with respect to pressure 55
Figure 2-35: The derived curve of losses at full displacement at the pressure of interest 55
Figure 2-36: Linear interpolation of losses with respect to displacement fraction 56
Figure 2-37: The derived curve of total losses at the pressure and displacement fraction of interest 56
Figure 2-38: The losses at zero speed and high pressure, due solely to leakage 57
Figure 2-39: Two cases for the value of k_{speed} 58
Figure 2-40: Two cases for the value of p_{zero} 59
Figure 2-41: The effect of the value of p_{zero} on the modelled leakage with respect to pressure 60
Figure 2-42: SD1B comparison between model (dashed) and experiment (solid); losses vs. speed; lines of pressure (bar) 62
Figure 2-43: SD1B comparison between model (dashed) and experiment (points); losses vs. pressure; lines of speed (rpm) 63

Figure 2-44: SD1B comparison between model (dashed) and experiment (points); leakage flow (l/min) vs. pressure (bar); lines of speed (rpm) 64
Figure 2-45: SD1B comparison between model (dashed) and experiment (solid); leakage flow (l/min) vs. speed (rpm); lines of pressure (bar) 65
Figure 2-46: SD1B comparison between model (dashed) and experiment (solid); torque loss vs. speed (rpm); lines of pressure (bar) 66
Figure 2-47: SD1B comparison between model (dashed) and experiment (solid); V_{eff} (cc) vs. speed (rpm); lines of pressure (bar) 67
Figure 2-48: SD1B comparison between model (dashed) and experiment (solid); overall efficiency vs. speed (rpm); lines of pressure (bar) 68
Figure 2-49: C2; Overall efficiency vs. speed (rpm) at full displacement; lines of pressure (bar) 70
Figure 2-50: C2; volumetric and mechanical efficiencies at full displacement, by standard definitions 71
Figure 2-51: C2; volumetric (dashed) and mechanical (solid) efficiency at full displacement, by revised definitions 72
Figure 2-52 C2 comparison between model (solid) and experiment (points); losses (W) vs. speed (rpm); lines of pressure (bar) and idle (purple) 74
Figure 2-53: C2 comparison between model (solid) and experiment (points); effective displacement (cc) as a function of speed (rpm); lines of pressure (bar) 75
Figure 2-54 C2 comparison between model (solid) and experiment (points); overall efficiency as a function of speed (rpm); lines of pressure (bar) 76
Figure 2-55: Comparing the C2 experiments (points) and model (dashed) with Dorey's model of an axial piston pump (solid); overall efficiency at full displacement vs. speed 77
Figure 2-56: Dorey's experimental data (points) and model (line) of the torque losses of a 161cc swashplate pump at 200 bar, 1500rpm 79
Figure 2-57: Alternative derivation of the relationship between displacement and torque losses in a swashplate pump; comparing Dorey's data to Dorey's model modified with a lookup table relating displacement to torque losses. 80
Figure 2-58: Comparison of the C2 experiments (points) and model (dashed) with Dorey's model, overall efficiency vs. displacement fraction, 1800rpm 81
Figure 2-59: Comparison of the C2 experiments (points) and model (dashed) with Dorey's model (modified), overall efficiency vs. displacement fraction, 1800rpm 81
Figure 2-60: Comparing the C2 model (solid) with the modified Dorey model of an axial piston pump (dashed) and manufacturers' data (points); losses (W) at 161cc scale vs. output flow (l/min); 1800rpm. 83
Figure 3-1: Basic DDPM operating cycles showing fluid flow paths 89
Figure 3-2: Overall timing diagram for the motoring cycle with NOSC LPVs and NCSO HPVs as defined below 91
Figure 3-3: Machine states during the DDPM motoring cycle (Coils: blue=off, red=on, pink=pulsed; Piston: red=high pressure, green=low pressure) 91
Figure 3-4: Choice of basic valve type for each of LPV and HPV, and possible cycles 93
Figure 3-5: Table of DD machine functionality by valve type 94
Figure 3-6: Solenoid formats according to Roters (1945) considered by Rampen (1992); (A) flat-faced armature (B) flat-faced plunger 96
Figure 3-7: Ideal one-dimensional model of solenoid valve motion 98
Figure 3-8: Ideal force vs. displacement (A) and force vs. current (B) characteristics 100
Figure 3-9: The 6 cylinder radial piston DDP by Rampen 102
Figure 3-10: Detail of a single DDP pumping element 102
Figure 3-11: Section of the baseline NCSO HPV design, designed by Rampen 103
Figure 3-12: Detail of the first generation NOSC LPV by Rampen, with adjustable return spring arrangement by the author 104
Figure 3-13 The first generation DDPM valves in the solid model assembly 104
Figure 3-14: Non-contact measurement of LPV poppet position with linear Hall effect sensor. Later the arm was replaced with a bracket, allowing measurements in a working machine. 105
Figure 3-15: The non-contact position sensing arrangement for HPV, complicated by the need to sense through a pressure bulkhead. 106
Figure 3-16: Definition of timing measurement for valve characterisation; trace from Mk.2 LPV for DDPM propel motor @ 14V in 32cst oil at room temperature. 107
Figure 3-17: Timing parameters for the same valve switching off from pulsed gate signal (22% duty cycle @ 10kHz) 107

Figure 3-18: System schematic for initial DDPM tests 109
Figure 3-19: The multi-cylinder DDPM, and the DDP used to supply it with fluid 110
Figure 3-20: FET pulses for the 6 cylinder DDPM motoring at 1500rpm 111
Figure 3-21: Coil drive circuit used in the DDPM 112
Figure 3-22: Coil drive equivalent circuits with a flywheel resistor: A=FET Gate on; B=FET Gate off 112
Figure 3-23: Annotated scope trace from early DDPM motoring cycle tests 114
Figure 3-24: Critical zones of timing adjustment for the DDPM 115
Figure 3-25: Effect on the HPV release time and pulse duty cycle with a low (A) and a high (B) value of flywheel resistance R_F 117
Figure 3-26: Effect of the ratio of the flyhweel resistance to the coil resistance 118
Figure 3-27: Prototyping the LPV poppet; (A) CAM planning; (B) Mill set-up; (C) Close up of machining; (D) The final part is compared to the solid model. Pictures courtesy of Jamie Taylor. 121
Figure 3-28: Design features of the DDPM-P LPV mk 1; (A) solenoid off, valve closed; (B) solenoid on, valve open (showing indicative flux lines) 122
Figure 3-29: Mk.2 LPV 123
Figure 3-30: Final version of LPV mk2; (A) assembled; (B) exploded 124
Figure 3-31: Matching the solenoid force curve (from FEA model) with the spring force 124
Figure 3-32: BH curve for EN1a mild steel 127
Figure 3-33: Design constraints for the HPV design improvement; left=baseline, right=improved 130
Figure 3-34: Parameterised geometry of HPV (final improved geometry shown) 133
Figure 3-35: Axi-symmetric geometry definition in ANSYS (Y is the axis of symmetry). Mesh coloured by material: red=air, green=steel, yellow=copper (coil), blue=non-magnetic stainless steel, pink=armature (also steel) 135
Figure 3-36: "Automatic" design optimisation method 137
Figure 3-37: An engineering approach to "optimisation" 139
Figure 3-38: Baseline design; "Grab" case 141
Figure 3-39: Baseline design; "Latch" case 141
Figure 3-40: Effect of coil internal diameter on grab force; 143
Figure 3-41: Effect of increasing the external diameter of the pole face 144
Figure 3-42: Concept of the flux concentrator 145
Figure 3-43: Results of parametric design improvement of the flux concentrator 146
Figure 3-44: Comparing the grab and latch cases with and without concentrator 147
Figure 3-45: The final armature geometry with minimised moving mass 148
Figure 3-46: Redesign of the poppet and armature of the improved HPV 148
Figure 3-47: Comparing latch force of baseline and improved HPV 150
3-48: Distinguishing between inductance L and incremental inductance L_{inc} 157
Figure 3-49: Simplified squeeze-film damping 158
Figure 3-50: Map of force vs. position and ampere-turns from FEA of the HPV 159
Figure 3-51: Solenoid force vs. ampere-turns in three armature positions; comparison of FEA results with the ideal model 160
Figure 3-52: Flux vs. position and ampere-turns from FEA 161
Figure 3-53: Incremental inductance; the gradient dAT of above 161
Figure 3-54: Step response of coil current vs. time for the LPV at a range of drive voltages 163
Figure 3-55: Step response of coil current vs. time for the HPV at a range of drive voltages 163
Figure 3-56: Table of HPV parameters for the transient model 164
Figure 3-57: Experimental results of current vs. time for the HPV, subject to transient excitation at a range of coil voltages. 165
Figure 3-58: Output of the hybrid FEA/LP model of the HPV, subject to transient excitation at a range of coil voltages 165
Figure 3-59: Comparison of the experimental results and the hybrid FEA/LP model 166
Figure 3-60: HPV total actuation delay 167
Figure 3-61: Simulation of HPV motion during DDPM cycle at 1800rpm; position. 168
Figure 3-62: : Simulation of HPV motion during DDPM cycle at 1800rpm; force 168
Figure 3-63: Spring and solenoid force 169
Figure 3-64: Model results of spring rate and preload on total actuation (t-total-a) and release (t-total-r) delays 170
Figure 4-1: DD secondary-controlled hydrostatic transmission concept 175
Figure 4-2: Section of the propel DDPM 177

Figure 4-3: Valves for the propel DDPM 178
Figure 4-4: Features of the radial-piston mechanism used in the propel DDPM. 178
Figure 4-5: Basic principle of a circular overclamped bearing 182
Figure 4-6: Articulation of the DDPM piston 183
Figure 4-7: The over-clamped hydrostatic bearings used in the fixed-speed DDPM; (A) detail of cylinder bearing; (B) cylinder solid model; (C) piston solid model 184
Figure 4-8: Catastrophic failure observed when starting on a hill; (A) piston and cylinder jammed between pump ring and eccentric; (B) impact damage 185
Figure 4-9: Stiction testing rig 186
Figure 4-10: Concept of the square piston pad 188
Figure 4-11: Stiction behaviour with the square pads; blue = stiction torque, pink = clamping ratio, red = qualitative leakage trend 189
Figure 4-12: Piston pad development. (A) original circular pad; (B) reduced land; (C) square pad; (D) square pad with reduced land; (E) solid model of first underclamped square pad as used in the propel DDPM demonstrator; (F) section of a later design with a screw-in impedance cartridge 192
Figure 4-13: Overview of the electronic system for the propel DDPM 194
Figure 4-14: Generation of FET gate signals with dedicated electronic circuit 196
Figure 4-15: Frames from a video showing the motor in the background, and the LED diagnostic tool in the foreground 197
Figure 4-16: Schematic of the system used for commissioning propel DDPM, also showing the later speed controller 199
Figure 4-17: Early propel DDPM development in the lab (from the video "Power for Change", courtesy Jamie Taylor) 200
Figure 4-18: Position measurement at start of false rev, going forwards 201
Figure 4-19: "False rev" sequence 202
Figure 4-20: Valve events for cylinder #6 as a function of shaft position 204
Figure 4-21: Valve event position advance as a function of speed 206
Figure 4-22: Valve events for the pumping cycle 207
Figure 4-23: Continuous displacement control of the propel DDPM 209
Figure 4-24: Secondary speed control of propel DDPM 210
Figure 4-25: Concept for secondary control of a propel DDPM in an electronic load-sensing system 212
Figure 4-26: Control algorithm for the propel DDPM vehicle 213
Figure 4-27: What can go wrong with an electronic load-sensing DD transmission 216
Figure 4-28: "Variable structure control" applied to DDPM speed control 218
Figure 4-29: The propel DDPM installed in the demonstrator vehicle, showing the absolute encoder, synchronised to the DDPM shaft with a small toothed belt 219
Figure 4-30: Layout of the demonstrator vehicle during development 219
Figure 4-31: Propel DDPM vehicle test results; torque control mode 223
Figure 4-32: Propel DDPM vehicle test results; speed control mode 225
Figure 4-33 Speed control mode in both directions 226
Figure 5-1: The converted DDP propel vehicle being demonstrated by the author to Dave Anderson, Sauer-Danfoss CEO, at the test track in Nordborg, Denmark 231
Figure 5-2: Overall system schematic 232
Figure 5-3: Table of important parameters 234
Figure 5-4: Solid model of the rear section of the golf buggy showing overall layout 235
Figure 5-5: Major system components 236
Figure 5-6: Plan view of buggy and section view of DDP mounted on engine 237
Figure 5-7: The golf buggy DDP, with tank tube removed 237
Figure 5-8: Design of the mounting of the Lucas motor 238
Figure 5-9: Close-up of the belt drive and tachogenerator arrangement 239
Figure 5-10: View from the back of the buggy showing displacement actuator and manifold block 239
Figure 5-11: Two-speed displacement actuator made for the Lucas motor 240
Figure 5-12: Simulation of flow output (normalised such that the peak flow from single full stroke = 1) of a 6 cylinder DDP; blue=flow output, red=displacement fraction demand; top=full strokes; middle=fixed part stroke (no mixing); bottom=variable part stroke 243
Figure 5-13: Detail of the low-flow region; top: blue=flow with only full strokes, green=flow with fixed part stroke (no mixing); bottom: blue=flow from variable part stroke 244
Figure 5-14: Calibration of the part stroke timing 246
Figure 5-15: Derivation of the LPV closing time delay 247

Figure 5-16: Effect of poor calibration on the flow linearity of a DDP with the fixed part stroke algorithm 248
Figure 5-17: Linearity of the (unloaded) motor speed to the DDP displacement demand 249
Figure 5-18: Displacement demand analogue interface; each point represents one step of the 8-bit DAC. 250
Figure 5-19: Run-time adjustable parameters in the system controller and an extract of the terminal log showing parameter 10 being adjusted 251
Figure 5-20:Format of the RS232 data collection and typical terminal log showing the data stream ..252
Figure 5-21: Simplified schematic of the system controller software 253
Figure 5-22 The driver interface for the propel demonstrator vehicle 254
Figure 5-23: Acceleration mode: control function 256
Figure 5-24: Acceleration mode: expected behaviour 256
Figure 5-25: Acceleration control algorithm 257
Figure 5-26: Position control algorithm 258
Figure 5-27: Simulation of the position control function, showing continuous acceleration with a step of handwheel rotation speed 259
Figure 5-28: Speed control algorithm 260
Figure 5-29: Test 1-3; measurement of engine speed by trigger pulse period 264
Figure 5-30: Test 1-1; RS232 data 266
Figure 5-31: Synchronising the DAQ and RS232 data 267
Figure 5-32: Detail of above figure 267
Figure 5-33: Test 1-3; pump pressure (red), motor speed (blue) 268
Figure 5-34: Test 1-3; Displacement demand voltage (green) and smoothed DDP current consumption(pink) 268
Figure 5-35: Test 1-3; Estimated pump torque (blue) and resulting filtered engine speed droop (red) 269
Figure 5-36: Test 1-3; correlation of calculated pump torque and engine droop 270
Figure 5-37: Test 8; effect of varying displacement change delay 271
Figure 5-38: Test 8; detail of change to low displacement; delay too long 271
Figure 5-39; Test 8; details of change to low displacement; delay correct 272
Figure 5-40: Test 2-4; Response of system to steps of DDP displacement 273
Figure 5-41: Test 2-4; actual speed and ideal speed (from displacement demand) 273
Figure 5-42: Test 2-4; motor speed period of oscillation 274
Figure 5-43: Test 2-3; effect of first-order filter; red=pressure, blue=actual speed, green=displacement demand. 274
Figure 5-44: Test 2-3; effect of first-order filter; blue=actual speed, red=speed demand 275
Figure 5-45: Test 2-5; symmetry of system response to speed demand steps 275
Figure 5-46: Test 2-6; step demand to full positive and negative speed demand; red=pressure, blue=motor speed 276
Figure 5-47: Test 2-6; software pressure limit in action 276
Figure 5-48: Test 6-2; 'jerk start' feature; red=pressure, blue=motor speed, green=pump displacement demand 277
Figure 5-49: Detail of 'jerk start'; red=pressure, light pink=unfiltered current, pink=filtered current, green=displacement demand 277
Figure 5-50: Test 7-1; blue=speed, red=pressure 278
Figure 5-51: Test 7-1; actual and demand position 279
Figure 5-52: Test 7-4; blue=speed, red=pressure 279
Figure 5-53: Test 7-4; actual and demand speed 280
Figure 5-54: Test 7-4; blue=actual position, red=demanded position 281
Figure 5-55: Test 7-4; blue= speed from motor tacho, green=derivative of measured position 281
Figure 5-56: Test 7-5; blue=speed, red=pressure 282
Figure 5-57: Test 7-5; blue=actual position, red= demanded position 282
Figure 5-58: Test 9-5; creeping with full strokes; blue=speed from motor tacho, red= pressure 284
Figure 5-59: Test 9-6; creeping with part strokes; blue=speed from motor tacho, red=pressure 284
Figure 5-60: Test 9-5; Detail of pressure and speed pulsation when creeping with full strokes; blue=speed from differentiating the position sensor, red=pressure 285
Figure 5-61: Test 9-6; Detail of pressure and speed pulsation when creeping with part strokes; blue=speed from differentiating the position sensor, red=pressure 285
Figure 5-62: Test 9-5; full stroke mode; red=acceleration from double differentiation of measured position, green=filtered accelerometer signal, pink=unfiltered current. 286

Figure 5-63: Test 9-6; part stroke mode; red=acceleration from double differentiation of measured position, pink=unfiltered current, green=filtered accelerometer signal 286
Figure 6-1: Top level of the Dymola model 295
Figure 6-2: The Control section of the model 296
Figure 6-3: The Engine section of the model 297
Figure 6-4: The Hydraulics section of the model 298
Figure 6-5: The overcentre valve model 300
Figure 6-6: Mechanics section 301
Figure 6-7 The simulated rolling resistance of the vehicle 302
Figure 6-8: Table of parameters describing linking of mechanical and hydraulic domains 303
Figure 6-9: Test used to estimate hydraulic system compliance and leakage 304
Figure 6-10: Experimental investigation of the compliance of the dead-headed vehicle hydraulic system, excited by full pumping strokes from the DDP at 1/3rd enabling fraction 305
Figure 6-11: Results from the dead-headed system compliance test, excited with nominal 17% part strokes, at close to full enabling fraction 306
Figure 6-12: Simple Dymola model of the DDP pumping into a closed volume with a relief valve ... 307
Figure 6-13: Identification of model parameters for system compliance and leakage - comparison of experimental pressure and current (black) and simulation (blue=pressure, red=DDP flow). 308
Figure 6-14: Detail of above figure in the "pump-up" phase 309
Figure 6-15: Icon for the DDP/Ideal pump model 311
Figure 6-16: Parameters for the DDP/Ideal pump model 312
Figure 6-17: Detail of the DDP/Ideal pump model 314
Figure 6-18: Icon of the detailed DDP model 315
Figure 6-19: The detailed DDP model 316
Figure 6-20: Calculation of the LPV closure angle from the desired part stroke fraction 317
Figure 6-21: The DDP flow algorithm 318
Figure 6-22: Generation of ideal flow pulses and the compressibility modification for each cylinder 318
Figure 6-23: Single cylinder flow simulation; traces numbered from the top. Trace 1: blue=normalised piston flow rate, red=pre-compressed flow rate, green=compressed flow rate; Trace 2: Pulse demand (1=pulse demanded); Trace 3:LPV state (1=closed); Trace 4: decrement pulse to flow algorithm 319
Figure 6-24: DDP/Ideal pump test model 320
Figure 6-25: Results from the DDP/Ideal pump test circuit with pump compressibility enabled; red=ideal pump, blue=Detailed DDP 321
Figure 6-26: Results from the DDP model with compressibility (1500rpm; 6 cylinders; 1/3rd enabling fraction) pumping into a closed dead volume with different part stroke fractions (pink=0.20, green=0.15, red=0.10, blue=0.05); Trace 1: pressure; Trace 2: flow rate filtered with first-order low pass (τ=0.05s); Trace 3: pressure zoomed-in 322
Figure 6-27: Effect of compressibility on flow linearity 323
Figure 6-28: Simulation results with ideal pump model; blue=simulation, red=experiment; from the top: pump pressure, motor speed, engine speed 325
Figure 6-29: Test 2-4; motor speed; blue=simulation, red=experiment 326
Figure 6-30: Test 2-4, response of vehicle speed to flow demand steps, with ideal pump and a range of system compliances. Black=0.4(CN=2.6), blue=0.8(CN=5.2), red=1.6(CN=10.5), green=3.2(CN=21), pink=6.4(CN=42) 327
Figure 6-31: Repeat of above with Test 2-3, with displacement demand filter 328
Figure 6-32: Test 7-4: position mode fast & loose, ideal pump; Top trace: vehicle speed: blue=simulation, red=experiment, green=ideal speed command; Bottom trace: blue=pump pressure (simulation), red=pump pressure (experiment), green=motor back-pressure from overcentre valve (simulation) 329
Figure 6-33: Test 1-1; Simulation results with detailed DDP model (blue=simulation, red=experiment); top trace=pump pressure; bottom trace=motor speed 331
Figure 6-34: Detail of area "a" (blue=simulation, red=experiment); top trace=pump pressure; bottom trace=motor speed 332
Figure 6-35: Detail of area "b" (blue=simulation, red=experiment); top trace=pump pressure; bottom trace=motor speed 333
Figure 6-36: Detail of area "c" (blue=simulation, red=experiment); top trace=pump pressure; bottom trace=motor speed 334

Figure 6-37: Test 1-1, comparison between simulation results with detailed DDP model (red), ideal pump model (blue) and the difference (green); top trace=pump pressure; middle trace=motor speed; bottom trace=vehicle acceleration 335
Figure 6-38: Test 7-1, ideal pump simulation (blue) and experiment (red); top trace= pressure, bottom trace=motor speed 336
Figure 6-39 Test 7-1, detailed DDP simulation (blue) and experiment (red); top trace= pressure, bottom trace=motor speed 336
Figure 6-40: Test 7-1, ideal pump simulation (blue), experiment (red), expected from displacement demand (green); top =vehicle position, bottom =vehicle speed 337
Figure 6-41: Test 7-1, detailed DDP simulation (blue), experiment (red), expected from displacement demand (green) ; top=vehicle position, bottom =vehicle speed 337
Figure 6-42: Simulated DDP flow for Test 7-1 (blue=flow, red=flow filtered with first order low-pass τ=0.05s); top=full strokes, middle= fixed part stroke (ps_fraction=0.166), bottom=variable part stroke 339
Figure 6-43: Zoom in on t=2s to t=4s of figure above 339
Figure 6-44: Test 7-1; simulation of the effect of the DDP flow algorithm: blue=full strokes, red=fixed part stroke (ps_fraction=0.166), green=variable part stroke; top=position, middle=speed, bottom=acceleration 340
Figure 6-45: Pressure trace for above tests 340
Figure 6-46: Effect of system compliance on vehicle behaviour with full strokes; blue:v=6.6l(CN=43.2), red:v=3.3l(CN=21.6); green:v=1.65l(CN=10.8); pink=ideal behaviour from displacement demand; top trace=position, middle trace=velocity, bottom trace=acceleration 342
Figure 6-47: Hysteresis and deadband in a swashplate pump with EDC 344
Figure 6-48: Test 7-1; comparison of swashplate pump with simulated and experimental DDP; blue=simulated DDP, red=experiment with DDP, green=simulated swashplate pump, pink=expected behaviour from and ideal system from displacement demand; top trace=position, bottom trace=speed 344
Figure 6-49: Displacement demand and actual swashplate displacement in above simulation 345
Figure 6-50: Simulated swashplate hysteresis loop in above simulation; x-axis=demand, y-axis=actual displacement fraction 345
Figure 9-1: Inputs and outputs of a forward-facing pump model 370
Figure 9-2: Inputs and outputs of a backward-facing pump model 371
Figure 9-3: Early attachment means for LPV 374
Figure 9-4: "Snap fit" concept of armature attachment for the LPV 375
Figure 9-5: Evolution of LPV poppet design; (A) Staked; (B) Snap fit; (C) Glued tube 375

1 Introduction

Digital Displacement (DD) machines are positive-displacement fluid pumps and pump/motors which are commutated by solenoid-actuated valves. Control of these valves by an embedded controller, sensing the shaft position, allows the machines to work as variable-displacement pumps or pump/motors, without requiring a mechanism to vary piston stroke.

Artemis Intelligent Power Ltd. (AIP) was formed by inventors Stephen Salter and Win Rampen in 1994 to commercialise DD technology. Since 1998 AIP has worked with Sauer-Danfoss Inc. to develop DD technology, and commercialise it for the mobile hydraulic market. This thesis documents some of the work undertaken by the author towards this goal since joining AIP in January 1999.

1.1 Review of DD fundamentals

The Digital Displacement Pump/Motor (DDPM) is a positive displacement fluid pump/motor commutated by solenoid-actuated poppet valves. Such machines have two ports: a high-pressure port, capable of providing or absorbing high-pressure fluid; and a low-pressure port which is connected to an atmospheric tank or a boosted supply of low-pressure fluid.

An embedded controller decides, on a stroke-by-stroke basis, whether a working chamber should execute a pumping, motoring or idling stroke and actuates the commutating solenoid valves accordingly. Modulation of the timing of commutating valve actuations allows the machine to deliver pumping or idling strokes which displace a fraction of the full volume of the working chamber. Control of fluid displacement of the machine is achieved by varying the time-averaged proportion of working chambers which execute partial (or full) pumping and motoring strokes, compared to those which execute idling strokes. Each high-pressure fluid pulse produced or absorbed by each working chamber is individually commanded by the controller.

A working chamber executing idling strokes is isolated from the high-pressure port, and thereby that working chamber mechanism is unloaded, causing no causing volumetric loss or pressure-related mechanical loss.

The Digital Displacement Pump (DDP) is a simpler variant of the DDPM with a passive high-pressure valve, in which each working chamber is capable of supplying, but not absorbing, high-pressure fluid.

Each working chamber may supply a separate hydraulic load or may be ganged with other working chambers to form a larger-capacity service. This allows a DD machine to supply multiple hydraulic loads with differing flow and pressure requirements.

It is recongnised that the DDPMs considered herein are not capable of reversing the functions of the low-pressure and high-pressure ports, and are therefore not direct replacements for all applications currently served by pump/motors commutated by valves or port plates mechanically connected to the shaft.

In terms of commercially available machines, a DDP is most similar to a radial-piston check-valve type pump commonly used for high-pressure industrial applications, but with the added advantage of computer-controlled variable displacement.

The DD principle has been applied to many different mechanical arrangements for the working chambers. Rampen's first DDP prototype was based on an axial-piston wobble-plate (Rampen 1992). However, since 1996, DD prototypes created by AIP have followed a radial-piston mechanical design by Rampen and Salter as shown below.

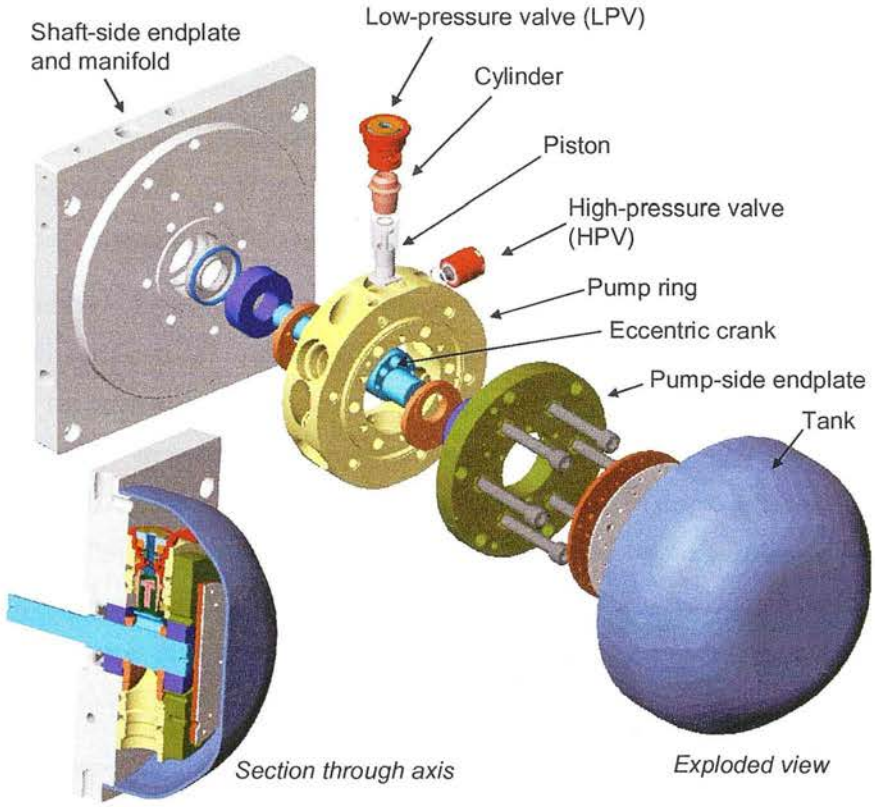


Figure 1-1: Radial section and exploded view of a radial-piston DDPM

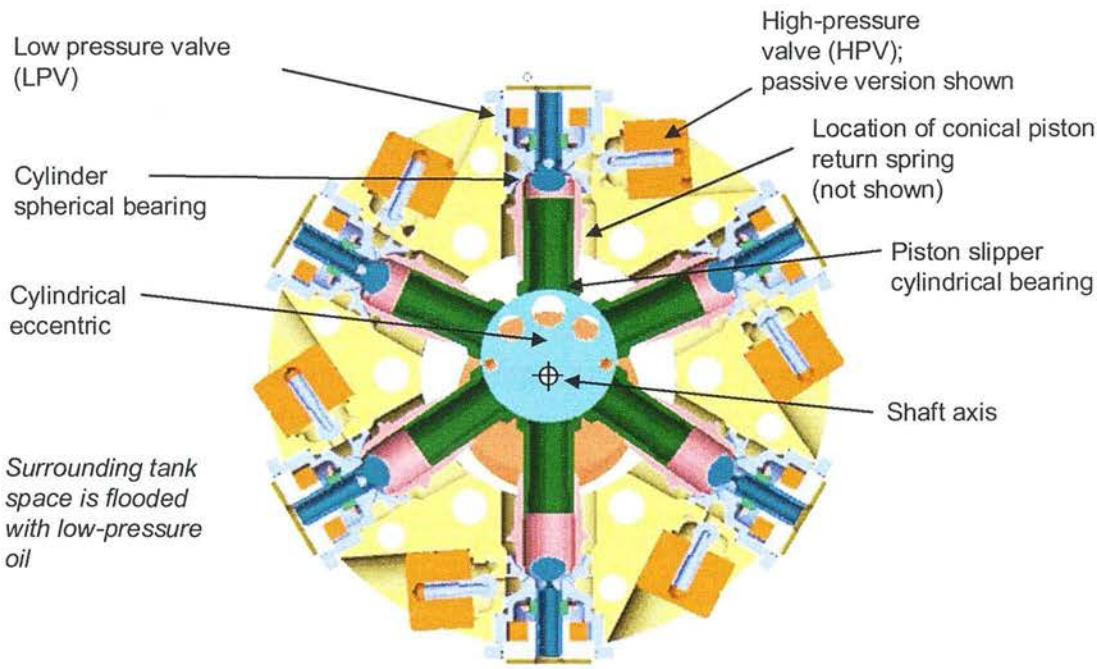


Figure 1-2: Section of a radial-piston DDP, normal to the axis

This radial-piston design was chosen by Rampen and Salter for the following key reasons:

- The shearing velocity of the highly-loaded piston slipper bearing is minimised, whilst the flow area available for the valves to supply fluid to the working chambers is maximised.
- Multiple banks of working chambers can be arranged axially on a common through-shaft.
- The self-aligning kinematic design of each working chamber relaxes geometrical tolerances of the structure, and allows the machine to be designed for strength rather than rigidity.

Further details on the DDPM mechanism and operation are given in subsequent chapters.

1.2 Literature on DD technology

The small body of published work which refers directly to DD technology is reviewed here, most of which originates from Salter, Rampen and thier collaborators. There is however, a wide body of work which intersects the field, without directly referring to the DD principle; reviews of such literature are contained in relevant chapters herein.

1.2.1 Early work

The Digital Displacement concept first makes an appearance in the academic literature in a paper by Rampen and Salter (1990). This describes the basic principle of a DDP, using an actively controlled inlet valve (also called the low pressure valve, or LPV) and passive check outlet valve (also called the high pressure valve, or HPV). By this time a patent had been granted to Rampen and Salter for the DDP (Patent No. EP0361927, 1990).

This paper discusses the control method of enabling cylinders on a stroke-by-stroke basis to satisfy a flow demand signal, and to close a pressure control loop using a demand signal and a pressure feedback signal. Results of initial trials with variable timing are discussed and the authors concluded that anything other than full pumping strokes would lead to unacceptable noise and mechanical shock. It was assumed that an accumulator would be used to smooth the pressure ripple caused by disabling complete cylinders, and some of the problems associated with accumulators in this role are discussed.

Results are presented from a simulation of a DDP with nine cylinders evenly spaced around a single eccentric. In the displacement control mode, the results of the steady-state pressure ripple at 50% displacement both with and without accumulator are presented. Here the ripple is periodic as might be expected, because every second cylinder is enabled. However, the results at 43% displacement show non-periodic (or pseudo-random) characteristics because the average number of cylinders enabled per revolution is not an integer, but is 3.87.

The response of a pressure control loop was investigated, with the DDP pumping at a constant demand pressure into an orifice load, which is subject to step changes of diameter. The response and recovery characteristics are shown to compare favourably with an axial piston pump controlled by a servo-valve, studied by Akers and Lin (1987).

The advantages over axial piston pumps claimed by Rampen and Salter are the tolerance of poppet valve machines to poor fluid quality, a potential for low noise due to the recovery of the compressibility energy at the end of each stroke, and energy efficiency improvements.

Most of two later papers (Rampen et al. 1991, Rampen et al. 1992) are contained within Rampen's PhD thesis, considered below.

1.2.2 Rampen's PhD Thesis (1992)

Rampen's PhD thesis (1992) is the largest single work on the subject. The inspiration for the invention was the requirement for a hydraulic power-takeoff scheme for the "Duck" wave power device. Salter and Clerk developed the idea of disabling working cylinders of positive displacement hydraulic machines under the command of a computer, as a way of reducing the parasitic losses from the pumps and motors when working at part-displacement.

Rampen first examined the fundamental control problem of the DDP, detailing the flow and pressure control algorithm and results of simulation.

Rampen considered from a fundamental level what form the active intake valves in a DDP should take. Before settling on the solenoid as the valve actuator, he examined other possible principles such as piezoelectric and magnetostrictive effects. He rejected both as being unable to provide the magnitude of travel required. It is interesting to note that since then, piezoelectric materials have been commercially developed for controlling high-pressure fluid valves at the extremely small scale of fuel delivery valves for common-rail diesel injectors (Ewing 2005). Although capable of only 40 μm of travel, such injector valves nevertheless show promise as the pilot stage for future spool valve designs (Reichart & Murrenhof 2006).

Rampen considers the issue of pressure drop through the disabled LPV, which is important because of an energy loss, and the force which must be resisted by the magnetic latch. From a series of experiments on the preferred elliptical profile, equations were derived showing how the pressure drop and force varied with with cylinder displacement, valve stroke, crankshaft speed, and fluid density and viscosity.

Rampen also considered some mechanical issues relating to making a demonstration DDP based on a commercial wobble-plate pump. Stress in the poppet material and fluid damping effects on the poppet valve were analysed.

The implementation of a controller based on the Intel Corp. 80196 microcontroller is described, from both hardware and software perspectives. This is the same microcontroller used by the author in the demonstration vehicles described later.

Results are presented recording performance tests on a DDP prototype, with particular emphasis on the pressure control loop.

1.2.3 The emergence of the DDPM

Salter and Rampen's patent (Patent No. EP0494236, 1992) extended the Digital Displacement concept to the hydraulic motor. A method is described for opening the high-pressure valve (HPV) at top dead centre (TDC), without requiring that the solenoid be strong enough to open against pressure; this method is described in detail in Section 3 herein.

Salter et al. (1993) described a design for such a machine on a megawatt scale, suitable for use as the power take-off system in a wave power device. Ehsan et al. (1995) considered the control aspects of a system based on this machine. The concept is discussed by Salter et al. (2002) in the context of the problem of power take-off from wave power converters.

Rampen et al. (1994) reported the results of first trials with a single-cylinder DDPM. Traces were shown of cylinder pressure and valve actuation signals during pumping and motoring cycles. The valve configuration for this machine is shown below:

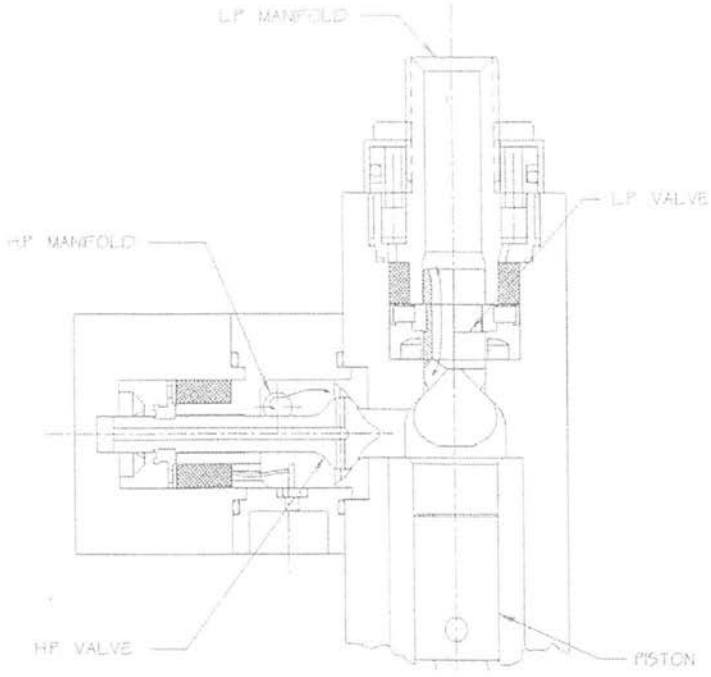


Figure 1-3: The single cylinder DDPM by Rampen et al., 1994. (courtesy Win Rampen)

By this time, the LPV had changed from the moving magnet type described in Rampen's 1992 thesis. Instead, the axially-polariased permanent magnet forms a stationary magnetic latch on the poppet side of the armature. Guidance of the armature comes from the poppet, allowing the radial gap to be large enough to dispel concern over fouling.

The HPV actuator followed a similar design but the desired direction of actuation and latching forces were reversed. The HPV was latched by a stationary permanent magnet latch in the open position, and closed by the solenoid. To achieve this, the poppet was actuated by a rod which passed through the coil. The armature was guided by the poppet as for the LPV. One drawback of this arrangement was that the coil of the HPV was exposed to the full cylinder pressure, so the sealing of the connections became critical.

1.2.4 Later work

Ehsan et al. (1996) described a computer model of a DDPM with the flow control mode extended to cover both pumping and motoring flow. Ehsan developed the model as part of his PhD research (Ehsan 1997) into the Malone heat engine (a variant of the Stirling engine). Other results of this work were also published (Rampen et al. 1997) and (Ehsan et al. 2002).

Hydraulic power take-off of the Malone engine requires the DDPM to take over the usual function of the crankshaft, but with variable amplitude, phase and motion profile. The DDPM produces an alternating flow in and out of a chamber which is pressurised by the heat engine. The net energy flow from the engine must be processed through the DDPM a number of times, so efficiency is crucial.

Uwe Stein built a working Malone engine with DDPM power take-off for his PhD thesis (Stein, 2002).

(Ehsan et al. 1997) developed an extension to the DDPM concept first proposed by Salter (1993). By connecting multiple DDPM units on a common shaft, the machines can be used together as a hydraulic transformer. This has possible application in injection moulding machines, where the transient power demands can be levelled by use of an accumulator as a short-term energy buffer. Later this concept was further developed (Ehsan et al. 2000, Ehsan and Rampen 2002). Ehsan's model is discussed in more detail in Chapter 6.

US Patents US2005073125, US2005085970, US2003102646, US2003168828, US2002195789, US2003001353 assigned to Visteon Global Tech Inc. relate to the application of a four-service DDPM to a vehicle chassis active body control system. Here, the DDPM acts as a "volume modulator" which displaces units of fluid into and out of a suspension strut in the form of a single-acting ram containing a compressible fluid. The regenerative ability of the DDPM is used to minimise power consumption, whilst the low idle loss minimises adverse effects on fuel consumption. These patents result from commercial cooperation between AIP and Visteon.

World patent application WO2006055978, assigned to Dana Corp., describes an on-road hydraulic hybrid vehicle system including DDPMs. This patent and others by Dana Corp. result from commercial cooperation with AIP. In 2006 rights to these patents were acquired by Bosch-Rexroth AG as part of its purchase of a license to DD technology from Dana Corp.

Payne et al.. (2006) present the application of a DDPM to the power take-off of a wave-energy device using the loss modelling method developed by the author, as described in Chapter 2.

1.3 Thesis

This work aims to establish the feasibility of the DDP and DDPM for application to vehicle propel transmissions. Specifically, the theses of this work are that:

1. The efficiency of the DDP can be well described with a semi-empirical mathematical model.
2. The DDP offers very significant energy efficiency advantages compared to a swashplate pump.
3. The performance of the DDPM is crucially dependent on optimising the response speed of the commutating valves, and magnetic finite element analysis is an effective tool to achieve this goal.
4. A DDPM can be made which is capable of starting from zero speed, and operating with variable displacement in all four quadrants.
5. A hydrostatic transmission comprising a DDP and a DDPM is capable of propelling a vehicle.
6. A hydrostatic transmission comprising a DDP and a conventional axial piston motor can be made which exhibits smooth motion control without perceptible pulsation.
7. The DDP offers control advantages compared to a swashplate pump, for vehicle propel transmissions.
8. Non-linear time-domain simulation is an effective tool for analysing such systems.

1.4 Chapter summary

The following is a summary of the work contained in the following chapters.

Chapter 2: Efficiency of the DDP

Literature relating to pump loss modelling is reviewed. Some observations are made about the special issues relating to measuring the efficiency of DDPs. Results are presented from tests on a 35cc/rev DDP. From the characteristics observed in these tests, a semi-empirical model is described and the results from this model compared to experiment. Results from an improved 12cc/rev machine are then presented, and a new set of modelling parameters established. The results from the improved machine are compared with an axial piston pump.

Chapter 3: Development of valves for the DDPM

Literature relating to solenoid valve analysis is reviewed. The basic functional requirements of valves in a DDPM are discussed. Results are presented from early trials with a DDPM showing that the magnetic circuit of the valves needed to be improved. Magnetostatic finite element analysis with parametric geometry was carried out, and the design investigated relating to its sensitivity to important geometrical parameters. The HPV was substantially improved by this method, leading to improved overall DDPM performance. A special active low pressure valve is described for the propel DDPM, the development and prototyping of which is also described. A hybrid FEA/lumped parameter model of valve motion is presented, and used to investigate valve behaviour.

Chapter 4: The DDPM propel demonstrator vehicle

Literature relating to secondary control of fluid motors is reviewed. The concept of the DDPM is reviewed and the required function of the valves is investigated. A method is described to allow a DDPM to start from rest. The mechanical and electronic design of the DDPM propel prototype is described, including the development of a hydrostatic bearing which reduces the friction of the piston pads. The motor control software is discussed. Results are presented from bench tests on secondary control of DDPM shaft speed. A demonstrator vehicle with a hydrostatic transmission system composed of a DDP and a DDPM is described, and experimental results are presented.

Chapter 5: The DDP propel demonstrator vehicle

A hydrostatic transmission system is described consisting of a DDP, a swashplate motor and ancillary valves. Implementation on a demonstrator vehicle is described with special attention to control algorithms. Experimental results are presented.

Chapter 6: Simulation of the DDP propel system

Literature relating to the simulation of hydraulic systems is reviewed. Time-domain models of the DDP and the propel system are described and the results are compared to the experiments described in the previous chapter.

1.5 Chronology

The work presented here took place between 1999 and 2006, but the order in which the chapters are presented differs from the actual order of the work. The work was undertaken as follows:

1999: A 6-cylinder fixed-speed DDPM was developed and constructed. Tests highlighted deficiencies in the valves prompting the investigation of the solenoid magnetics as described in Chapter 3.

2000-2001: A 6-cylinder DDPM was developed and constructed which was capable of starting from zero speed and therefore suitable for propelling a vehicle. This DDPM was integrated into a hydrostatic transmission supplied with fluid from a DDP then installed in a demonstrator vehicle based on a golf buggy, as described in Chapter 4.

2002: The demonstrator vehicle was converted to demonstrate a hydrostatic transmission comprising a DDP, valves and a conventional motor, as described in Chapter 5.

2004: Efficiency tests were undertaken at Sauer-Danfoss Inc. of a 35cc/rev DDP designed for mobile hydraulics application. Results from these and other tests are analysed in Chapter 2.

2005: A mathematical model of the efficiency of a DDP was developed and compared with experimental data, the results from which are presented in Chapter 2.

2006: A time-domain simulation model was developed of the DDP propel system, as presented in Chapter 6.

2 Efficiency of the DDP

2.1 Introduction

Energy efficiency was the main motivator to the conception of DD technology. In the mid-80's, Stephen Salter was searching for a mechanical power transmission for the "Duck" wave power generator (Salter et al. 2002). The input to this transmission was the oscillating motion of a structure excited by sea waves of variable amplitude and frequency, whilst the output was coupled to the shaft of an electrical generator turning at synchronous speed – typically 1500rpm. Salter concluded that only hydraulics could provide the required combination of high power capacity and computer-controlled infinitely-variable transmission ratio. To make the best economic return on the investment in the structure, the conversion mechanism had to be sized to capture the majority of energy when input power was high (which was infrequent), but also had to be efficient at low power levels (where it operated most of the time). However Salter's survey of commercially available hydraulic machines revealed that their part-load efficiency was not acceptable for this application.

Salter realised that variable-stroke positive-displacement machines have a fundamental problem at low fractions of full displacement – the losses due to leakage, compressibility and bearing friction are still high but the throughput power is reduced with the displacement, causing poor overall efficiency. One solution was to create an hydraulic machine in which displacement was varied by dynamically disabling cylinders instead of reducing the stroke. Disabled cylinders would not be pressurised, and so would attract no leakage, compressibility or pressure-dependent mechanical loss. As the displacement of the machine was reduced, most of the losses would reduce, keeping efficiency high even at small fractions of full displacement.

The desire for high efficiency at partial displacement is also common to many mobile hydraulic applications. In hydrostatic transmissions designed for a wide ratio spread e.g. wheel loaders, either the pump or the motor is at partial displacement most of the time, and the overall transmission efficiency with variable-stroke piston machines is typically between 60% and 80%, depending on load (Bowns et al. 1973). As fuel prices rise and engine emission regulations become more stringent, increasing the transmission efficiency becomes more important to the machine manufacturer and the end-user.

A significant fraction of the losses in a hydrostatic transmission comes from the pump. This chapter seeks to investigate the efficiency of the DDP, present experimental results and a mathematical model which matches them, and compare these with results from the commonly-used axial piston pump.

2.2 Literature review of loss models of hydrostatic machines

The manufacturers of mobile hydraulic machines need to be able to calculate the energy losses over typical duty cycles. Normally such work would be done by analysing maps of volumetric and mechanical efficiency as a function of speed and pressure, as this is the form that manufacturers supply data about their components. However this approach may not be suitable when applied to typical duty cycles involving variable-displacement machines:

- By definition the efficiency of a pump at zero output is zero; efficiency maps becomes increasingly inaccurate as the machine approaches the zero displacement or zero pressure condition, because power output approaches zero. The torque required to turn the shaft of a pump at such an idle condition cannot be calculated from an efficiency map.
- Manufacturers rarely publish efficiency maps of their machines at anything other than full displacement. In real duty cycles, either pump or motor or both are at fractional displacement most of the time (Bowns et al. 1973).

For duty cycle analyses, it is therefore preferable to be able to describe the absolute energy losses in a hydraulic component as a function of speed, pressure and output flow, rather than use efficiency maps. Most of the published academic models of the energy flow in hydrostatic machines deal in losses, rather than efficiency.

Typically, an electronics engineer choosing a transistor can download the important simulation parameters for a SPICE model from manufacturers' websites (Fairchild 1999). To the author's knowledge, no manufacturer of hydrostatic machines publishes models and model parameters for their products, but this has long been an active field of research in academia. Unfortunately, due to the complexity of the behaviour of hydrostatic machines, there are almost as many loss models as there are authors in the field.

Broadly there are two type of loss model in the literature:

Physical-modelling approaches, where the model structure reflects an analysis of the loss sources from physical principles and therefore includes assumptions about how the machine should behave;

Empirical approaches, where the model structure is created purely to match observed losses from experimental data.

2.2.1 Physical modelling approaches

McCandlish & Dorey (1984) and Huhtala & Vilenius (1997) present comprehensive surveys of physical modelling approaches. Much work has been based on the lumped-parameter analytical model by Wilson (1948). As later developed by others to apply to variable pumps with compressibility effects (Dorey 1988), the flow model for a variable-displacement pump is:

$$Q = x \cdot \omega \cdot D \cdot \left[1 - \frac{C_s}{x} \left(\frac{p}{\mu \cdot \omega} \right) - \frac{p}{x \cdot \beta} \left(V_r + \frac{1+x}{2} \right) \right] \tag{1}$$

and the torque model is:

$$T = x \cdot p \cdot D \cdot \left[1 + \frac{C_v}{x} \left(\frac{\mu \cdot \omega}{p} \right) + \frac{C_f}{x} \right] + T_E \tag{2}$$

where:

x is the fraction of full displacement at which the pump operates (between 0 and 1)

ω is the speed of the shaft ($rads^{-1}$)

μ is the fluid dynamic viscosity ($N\ m^{-1}\ s^{-1}$)

D is the displacement of the pump (m^3/rad)

β is the fluid bulk modulus (Pa)

p is the pressure at the pump outlet (Pa)

C_s is the slip coefficient, which models leakage.

V_r is the ratio between dead volume in the chamber and the swept volume.

C_v is the viscous loss coefficient, which models shear in thin oil gaps not related to load.

C_f is the Coulomb friction coefficient, which models load-related and speed-independent mechanical loss.

and T_E is a constant torque loss which models seal friction etc.(often omitted).

The form of the developed Wilson model can be compared to the expected behaviour of a DDP using cylinder-disabling as the method of changing displacement. Considering the development of Wilson's flow model as described by Dorey (1988):

- The leakage is proportional to pressure and constant with respect to both speed and displacement fraction. By contrast, the leakage in the DDP is expected to reduce as displacement fraction reduces because, on average, fewer working chambers are exposed to pressure.
- At zero displacement, the flow lost to compressibility is reduced by, at most, one half compared to full displacement. By contrast, the compressed volume for active cylinders is always the same with the DDP, so as displacement reduces the flow reduction due to compressibility should reduce in exact proportion. At zero displacement, the working chambers of the DDP are not pressurised so no flow can be lost to compressibility.
- It is assumed that all of the energy in the compressed chamber is lost. By contrast, check-valve pumps (like the DDP) can recover some portion of the compressibility energy after TDC, because the chamber is only connected to low pressure once the chamber volume has expanded and the chamber pressure has fallen to intake pressure, returning some fraction of the compressibility energy to the shaft (Edge & Brett 1990).

Considering the development of Wilson's torque model described by Dorey (1988):

- Both the friction and shear torque losses are constant with respect to displacement fraction. In the DDP however, as displacement fraction reduces, the load-related torque losses of the DDP should reduce in proportion, because on average, a smaller proportion of the working chambers is under load.

It is therefore obvious that the development of Wilson's model of a swashplate pump is not applicable to the DDP.

One of the attractions of the development of Wilson's model is that a particular design can be characterised by the four normalised coefficients, which are easy to calculate from experimental data. However, McCandlish and Dorey (1984) note that later work has shown that the development of Wilson's model with constant coefficients does not explain observed behaviour at all operating conditions.

To address these issues, Dorey (1988) proposes that for a piston pump:

C_s varies as a quadratic function of speed and an inverse function of pressure;

C_v is a linear function of displacement fraction;

C_f is a quadratic function of speed and a linear function of displacement fraction.

Using this form, Dorey achieves a good match with experimental results for a 161cc swashplate pump. Unusually, Dorey published both his model and the parameters for a particular machine, allowing his model to be re-created; most other researchers publish the form of their model but not the value of the parameters which they arrived at to match the experimental data. This model is used for comparison between DDP and swashplate pumps later in this chapter.

The inclusion of viscosity in Wilson's basic model outside of the C_s and C_v coefficients implies that the model can be used at a different viscosity to that used to calculate the coefficients. In fact Dorey notes that a Wilson-based model cannot be expected to model behaviour at any other viscosity than that used for the experimental results, for which the coefficients were calculated. This is perhaps to be expected because the shearing and leaking gaps of the hydrodynamic/hydrostatic bearings of the piston pads and the port plate are sensitive to viscosity changes. It may have been better to acknowledge this and remove viscosity from the equations altogether, dealing with any viscosity changes by interpolating between separate sets of coefficients provided for discrete viscosities.

Dorey includes speed and pressure within his new coefficients, when already they appear outside of them in the original Wilson model. Dorey has not therefore so much modified the Wilson model but created a new one, because Wilson's original assumptions have been invalidated.

Perhaps the most comprehensive of the physical modelling approaches is that by Bavendiek (1987) as reviewed by Ortwig (2003). Bavendiek analysed the loss sources in a variable-displacement bent-axis pump. Volumetric and torque losses were dealt with separately as a weighted sum of equations, each of which characterised a particular form of loss. There were thirteen torque loss equations and eight volumetric flow loss equations, each with associated weighting coefficients; it is notable that much of complexity of the model relates to the effect of the control angle. These coefficients were derived from experimental results by solving a system of equations.

Ortwig used Bavendiek's model to create a commercial software package to model hydrostatic transmissions. Therefore his observations about the utility of this approach are worth empasising:

> "...it emerges that the individual loss values are highly inaccurate and that only the aggregate loss value represents a useful quantity."

At this point Ortwig disassociated himself with the physical modelling approach and went on to describe an approach based on empirical curve-fitting.

It is the author's view that coefficients for physical-modelling approaches would ideally be arrived at by a series of individual investigations on special tests rigs acertaining the losses from each source in isolation – e.g. the leakage from a single reciprocating piston. This would certainly provide the most useful information for a machine designer interested in optimising the efficiency of components. Given the expense and practical difficulty of this method, it is no surprise that the majority of researchers prefer not to take machines apart. However it appears that without individual experiments on the components of a machine in isolation, the determination of physical-modelling coefficients from experimental results effectively amounts to a form of empirical curve-fitting.

2.2.2 Empirical approaches

Conrad et al. (1990) present a method for fitting the observed flow and torque of fixed-displacement pumps and motors with polynomial approximations, where the speed and pressure were independent variables. An advanced dynamometer rig is described that provided automatic acquisition of the required data.

Ortwig (2003) presents a method of approximating the observed behaviour of a variable-displacement hydrostatic machine using regression and interpolation methods. The extra degree of freedom of swashplate angle greatly complicates the task.

Huhtala et al. (1995) present a semi-empirical model based on extrapolating between curves collected at the extremes of pressure and speed.

Mikeska and Ivantysynova (2002) apply a similar method to the problem of power-split transmissions for tractors including bent-axis machines in the hydrostatic path. The "POLYMOD" model which is used creates separate descriptions of the volumetric and torque loss, both being general polynomial functions of speed, pressure and derived displacement (itself a function of control angle). Results are valid for only one viscosity; the extension of the model to include viscosity effects is possible only with "enormous measurement effort".

Recently a study has been made of a hydrostatic power take-off for the Pelamis wave-power device using DD machines using the modelling method developed by the author which is presented in this chapter (Payne et al. 2006).

2.3 Background to DDP loss models

Salter and Rampen modelled the losses in a DDPM using MathCAD software, as part of a design of a wave-power conversion mechanism (most of this work has not been published). The purpose of this model was to guide initial design of a radial-piston DDPM such that the machine had the lowest possible losses. This was a physical-modelling approach, the losses

from each component being calculated from theoretical analysis of geometry and material properties.

As the author and colleagues at AIP worked to develop DDP hardware, special test rigs were created to measure the actual losses from most of these sources. In some cases the actual behaviour differed from the analytical predictions due to simplifying assumptions inevitable in such analysis. Attempts were made to account for observed effects by replacing the analytical calculations with the empirical observations. For instance, the pressure drop through the disabled LPV differed substantially from one-dimensional analysis and therefore the analytical result was overridden with a polynomial function derived from experimental results. In this way the model evolved to a semi-empirical one, but still based on summing the losses from all the component sources.

As investigations proceeded into the possible benefits of DD technology on the efficiency of vehicle hydrostatic transmissions, it became important to be able to simulate the fuel consumption of such a vehicle over a complete drive cycle, so a loss model was needed which could be incorporated into a vehicle model.

The semi-empirical analytical model was too complex to be embedded into a vehicle simulation, and Rampen suggested a simplification by fitting polynomial curves to the predictions of the semi-empirical analytical model, similar to the approach taken by Ortwig and Huhtala to match experimental results. The loss sources were lumped together, with the analytical model as the input data rather than experimental results. At this stage there were few experimental efficiency results available with which to calibrate the model.

As experimental results became available, the author developed this lumped parameter loss model to match these results, with the aim of creating a model useful for system simulation. The description of these results and the model developed to match them forms the core of this chapter.

2.4 Analysis of DDP test results

This section describes the analysis of the results from the DDP efficiency tests.

Measuring the efficiency of a DDP is complicated by the high levels of torque and flow pulsation compared to other hydrostatic pumps, and special care must be taken with the the hydraulic circuit layout, instrumentation, and data analysis to ensure that the results are accurate. Appendix. 9.2 describes these practical issues and suggests methods for minimising their effect on the accuracy of the results; in the following analysis it is assumed that the effect of pulsation on the accuracy of data has been reduced to a negligible level.

Consider the case of a variable displacement positive displacement pump. The input shaft of the pump turns with speed ω (radians per second) and the pump applies torque T (Nm) to the shaft. The output flow is Q (m^3/s) and the pressure above atmosphere is p (Pa). It is assumed that the fluid inlet is at atmospheric pressure.

The output power is:

$$P_{out} = p \cdot Q \tag{3}$$

while the input power is:

$$P_{in} = T \cdot \omega \tag{4}$$

The overall efficiency of the pump is:

$$\eta_{overall} = \frac{P_{out}}{P_{in}} \tag{5}$$

If P_{loss} is the power lost inside the pump, this can also be expressed as:

$$\eta_{overall} = \frac{P_{out}}{P_{out} + P_{loss}} \tag{6}$$

Conventionally the overall efficiency of the pump is split into volumetric and mechanical efficiency:

$$\eta_{overall} = \eta_{vol} \cdot \eta_{mech} \tag{7}$$

And power losses are split into mechanical and volumetric:

$$P_{loss} = P_{lossmech} + P_{lossvol} \tag{8}$$

which, if mechanical losses are expressed as a lost torque, T_{loss} and volumetric losses are expressed as a flow Q_{loss}:

$$P_{loss} = \omega \cdot T_{loss} + Q_{loss} \cdot p \tag{9}$$

If V_{geom} is the geometric displacement of the pump (m^3/rad), then a simple way to calculate η_{vol} is:

$$\eta_{vol} = \frac{Q}{\omega \cdot V_{geom}} \tag{10}$$

However, due to errors in commutation timing and incomplete inlet fill, the maximum possible flow from a pump at atmospheric outlet pressure may deviate from the geometric displacement, as a function of speed.

ANSI/NFPA standard T.3.9.17 defines a method to take these effects into account, based on an experiment whereby at a range of speeds, the flow from the pump is recorded as a function of pressure. The extrapolated flow at zero pressure differential at this speed is termed Q_{th}.

The displacement arrived at by this method is termed the "derived capacity" V_i which is a function of speed:

$$V_i(n) = \frac{Q_{th}(n)}{\omega} \tag{11}$$

As pressure is applied to the pump outlet, the measured flow reduces due to the effects of compressibility and leakage. At a given pressure and speed, the pump will have an effective displacement V_{eff}, which is smaller than V_i:

$$V_{eff} = \frac{Q}{\omega} \tag{12}$$

Any deviation from Q_{th} at the measured pressure is assumed in ANSI/NFPA T.3.9.17 to represent energy lost, either as leakage or as compressibility. This is a common assumption in the analysis of positive-displacement pumps (Dorey 1988). While, as Dorey notes, this is almost always true of axial piston pumps commutated by port plates with fixed timing, it is not true of machines commutated by self-acting poppet valves (Edge et al.. 1990, Johnston 1991). In such machines, after TDC the delivery valve closes, and the oil expands until the pressure in the pumping chamber returns to the inlet pressure. During this expansion some portion of the energy contained in the compressed oil is returned to the shaft as work. Hence the common assumption that any deviation of Q from Q_{th} represents energy lost, is not appropriate for DDPs, which are a special class of self-acting poppet-valve pump. Leakage is clearly energy loss, but compressibility merely causes the displacement of the pump to reduce as the pressure increases.

In the analyses of pumps commutated with self-acting valves by Edge et al. and Johnston, the term "volumetric efficiency" is used to describe the ratio between the effective displacement and the geometric displacement. In the author's view this definition is misleading. Given that overall efficiency is defined the product of volumetric and mechanical efficiency, then the use of this definition for a pump with self-acting valves can lead to the unsatisfactory conclusion that mechanical "efficiency" has to be greater than unity.

In the author's analysis, volumetric efficiency contains only leakage losses, not the reduction in displacement due to compressibility. That is not to say that it is assumed that all compressibility energy is recovered after TDC; rather that the energy which is lost in this way

is allocated to mechanical loss, rather than volumetric loss. In this way it is guaranteed that both mechanical and volumetric efficiency are less than unity.

In axial-piston machines there is often a case drain connection from which the majority of the leakage emerges where it can be measured (external leakage). There is also some leakage across the ports of the machine (internal leakage) which cannot be measured directly. None of the leakage flow can be measured directly in any of the DDP prototypes considered because it returns to DDP case, which is internally connected to the pump inlet. Effectively, all of the leakage is "internal" and immeasurable; it must therefore be inferred.

The flow which is "missing" from the outlet of the pump, from either leakage or compressibility, can be termed Q_{miss}:

$$Q_{miss}(\omega, p) = Q_{th}(\omega) - Q(\omega, p) \tag{13}$$

where

$$Q_{miss}(\omega, p) = Q_{leak}(\omega, p) + Q_{comp}(\omega, p) \tag{14}$$

Dividing by ω yields the "missing" displacement:

$$V_{miss}(\omega, p) = V_{leak}(\omega, p) + V_{comp}(\omega, p) \tag{15}$$

V_{comp} represents the reduction in the effective displacement of the pump due to the compressibility effect. If a constant bulk modulus is assumed, V_{comp} is the first order of pressure and independent of speed. Therefore one possible method to arrive at Q_{comp} would be to examine the Q_{miss} function and split off that portion which is first order with respect to speed. However this cannot be achieved with any certainty as is illustrated below:

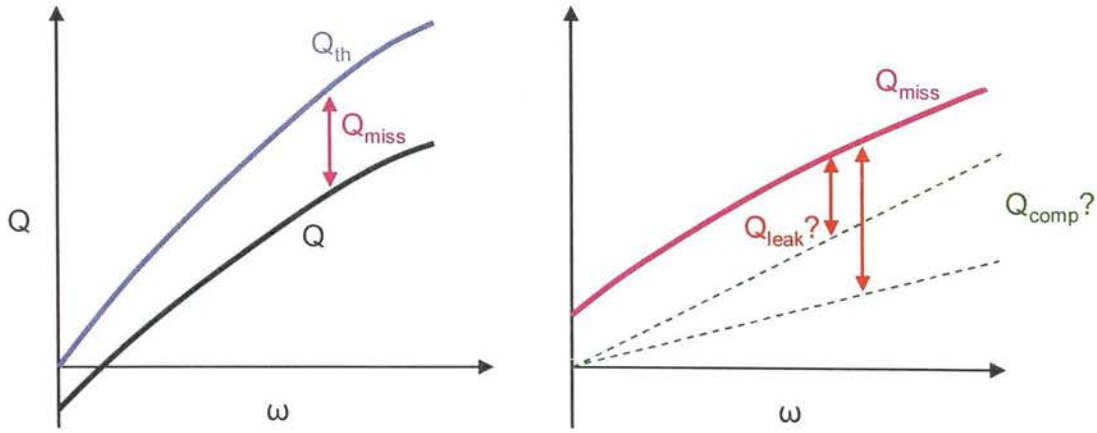


Figure 2-1: Uncertainty in the split of "missing flow" between the effects of leakage and compressibility.

Nothing can be assumed about the shape of the Q_{leak} function, because of the complexity of leakage behaviour of hydrostatic bearings as a function of speed and pressure. Hence the split of Q_{miss} is arbitrary without knowledge of the actual compressibility effect of the pump under test. As noted by Dorey (1988), this is a function of the ratio between the swept volume V_{geom} to the "dead" volume V_{dead} – that portion of the total pressurised volume inside all the pumping chambers which is not swept. A ratio k_{dead} can be defined:

$$k_{dead} = \frac{V_{dead}}{V_{geom}} \tag{16}$$

So the compressed displacement of the pump can be calculated:

$$V_{comp}(p) = V_{geom} \cdot (1 + k_{dead}) \cdot \frac{p}{\beta} \tag{17}$$

where β is the fluid bulk modulus.

So to arrive at a meaningful split of losses in a DDP into volumetric and mechanical losses, it is necessary to inspect the geometry of the pump under test to calculate k_{dead}.

Once k_{dead} has been calculated, Q_{leak} can be derived from the missing flow Q_{miss} by use of eq. 14. This allows the calculation of $P_{lossvol}$, and therefore $P_{lossmech}$, and therefore T_{loss}.

$$T_{loss}(\omega, p) = \frac{P_{loss}(\omega, p) - \left(Q_{leak}(\omega, p) \cdot p\right)}{\omega} \tag{18}$$

If a pump is 100% mechanically efficient, the torque exerted on the shaft is a function of the volume displaced by the pump (either into the load or into leakage) and pressure:

$$T_{theory}(\omega, p) = \left(V_i(\omega) - V_{comp}(p)\right) \cdot p \tag{19}$$

The mechanical efficiency is:

$$\eta_{mech}(\omega, p) = \frac{T(\omega, p)}{T_{theory}(\omega, p)} \tag{20}$$

and the volumetric efficiency follows from eq. 7.

Alternatively the volumetric efficiency η_{vol} can be calculated directly:

$$\eta_{vol}(\omega, p) = \frac{V_{eff}(\omega, p)}{V_i(\omega) - V_{comp}(p)} \tag{21}$$

and η_{mech} follows from eq. 7.

2.5 The SD1B test machine

The first high-quality efficiency data for DDPs were collected in 2004 at a Sauer Danfoss test facility as part of their assessment of the potential of DD technology. The machine tested was a 35cc/rev DDP, codenamed "SD1B", designed by AIP to be a "drop-in" replacement for an existing load-sensing open-circuit axial-piston pump in a mobile application. To achieve the required displacement, three banks were used, each bank comprising six cylinders of 1.94cc displacement each. Three cylinders from each bank were ganged together to form six individual services; these six services were themselves ganged with an external "function manifold" to create one service of 35cc/rev.

The design is shown in the following figures.

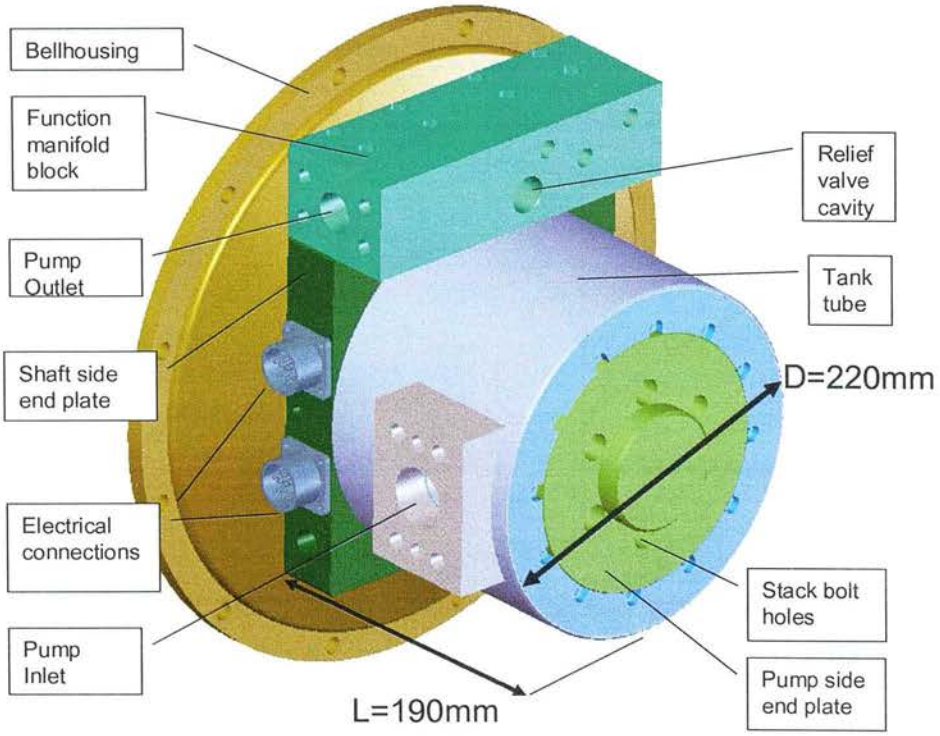


Figure 2-2: Isometric view of the SD1B overall design

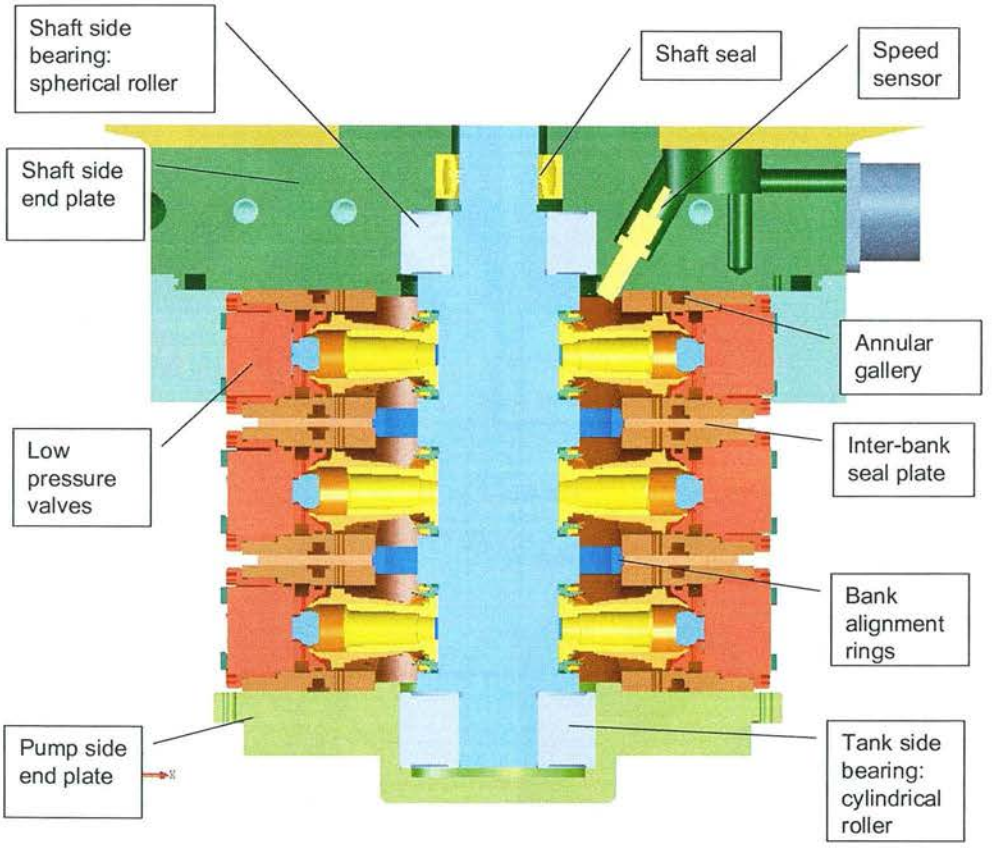


Figure 2-3: Section view showing three-bank design of the SD1B

The figure below shows the path taken by the fluid in the working chambers of the SD1B:

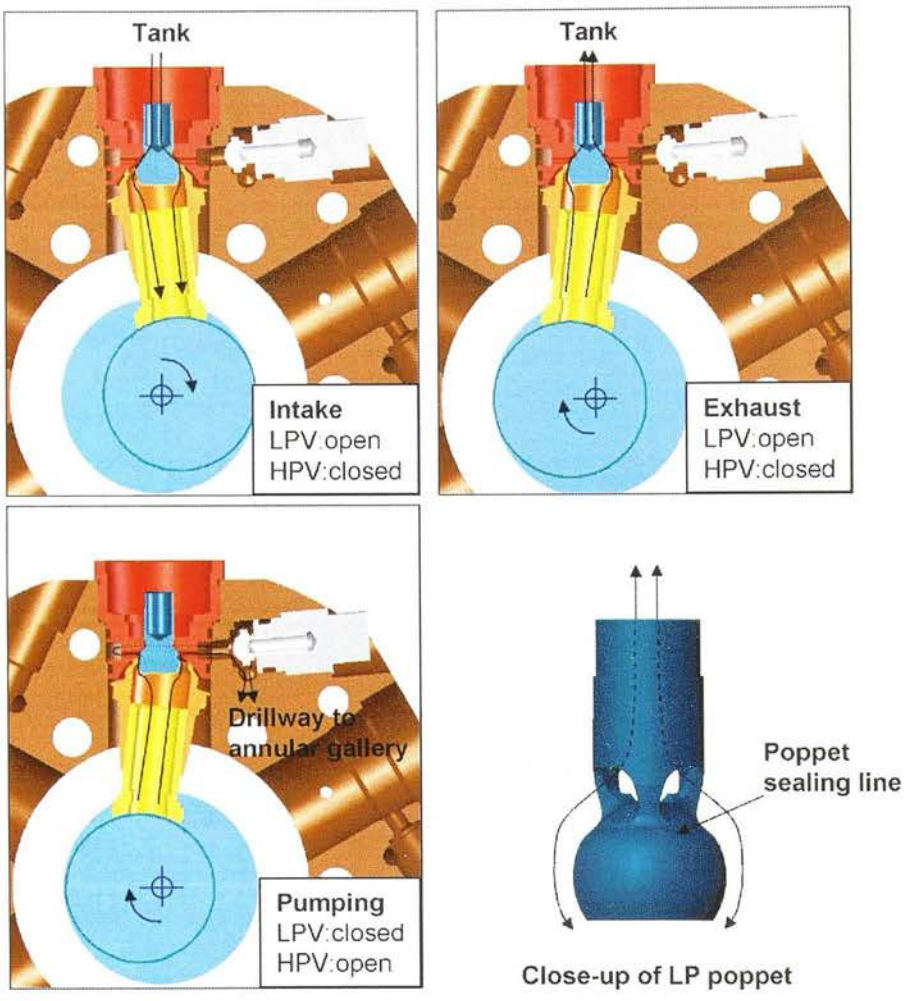


Figure 2-4: Flow lines in a single SD1B working chamber

A table of important parameters of the SD1B is shown below:

Overall length	190mm
Tank outside diameter	220mm
Mass	34kg
Geometric Displacement	34.9cc/rev
Number of cylinders	18
Cylinder diameter	15.0mm
Piston stroke	11.0mm
Porting configuration	6 services, each of 3 cylinders at 120° phase separation

Maximum speed	3200rpm
Maximum pressure	300 bar
Controller	Infineon C167 microcontroller
Control interface	RS232 terminal
Shaft speed sensor resolution	6 pulses per rev., plus 1 sync pulse
Controller analogue inputs	Pump pressure sensor Load sense pressure sensor Case temperature sensor Flow demand signal
Solenoid voltage	12V nominal
Solenoid pulse length	3ms duration
Electrical power consumption	1.7W per l/min

Figure 2-5: Table of important design parameters of the SD1B

The LS hose from the proportional valves was terminated with a pressure transducer and that signal fed to the controller, which also sampled pump outlet pressure from a second transducer. The error signal which resulted was then processed to create a displacement demand of the DDP such that the pump pressure matched the LS feedback line, plus a margin pressure

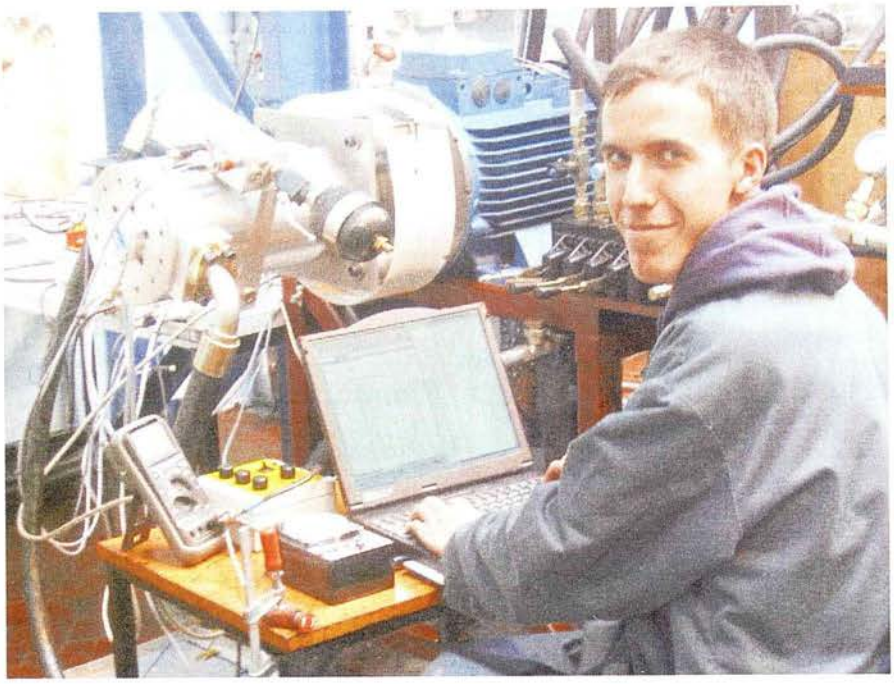


Figure 2-6: Pierre Joly calibrating the SD1B controller in the AIP lab

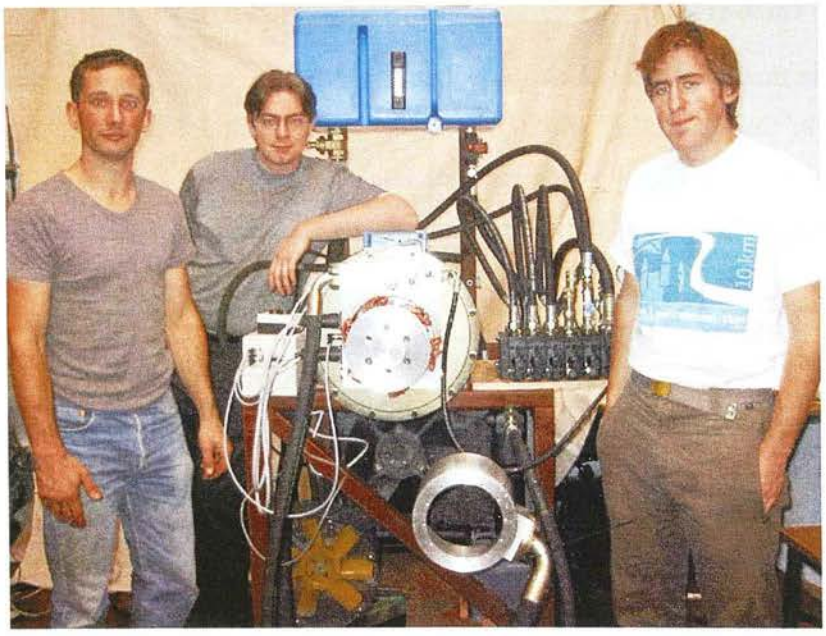


Figure 2-7: The SD1B being commissioned as an electronic load-sensing pump at the AIP lab in November 2003. From left to right: Uwe Stein, the author, Jack Lavender

Once the electronic load-sensing control function was implemented in the controller, the SD1B was installed in a mobile aerial work platform (Pingeuly-Haulotte HA16PX) in which both propel and working functions were supplied by a single load-sensing open-circuit pump.

Figure 2-8: Tests on an aerial lift fitted with the SD1B working as an electronic load-sensing open-circuit pump

These tests showed that an electronic load-sensing DDP could be successfully integrated into a mobile application. Pressure control performance was similar to the axial-piston pump which was replaced, and the vehicle behaviour was identical. Although there was significant pressure pulsation at the DDP outlet, there was no detectable increase of structural vibration or noise to the operator; this was due to the filtering action of the proportional valves between the pump and the actuators.

After the application study, the SD1B was taken to the Sauer Danfoss test facility in Ames (USA) for tests of efficiency and performance.

2.5.1 The test rig for the SD1B tests

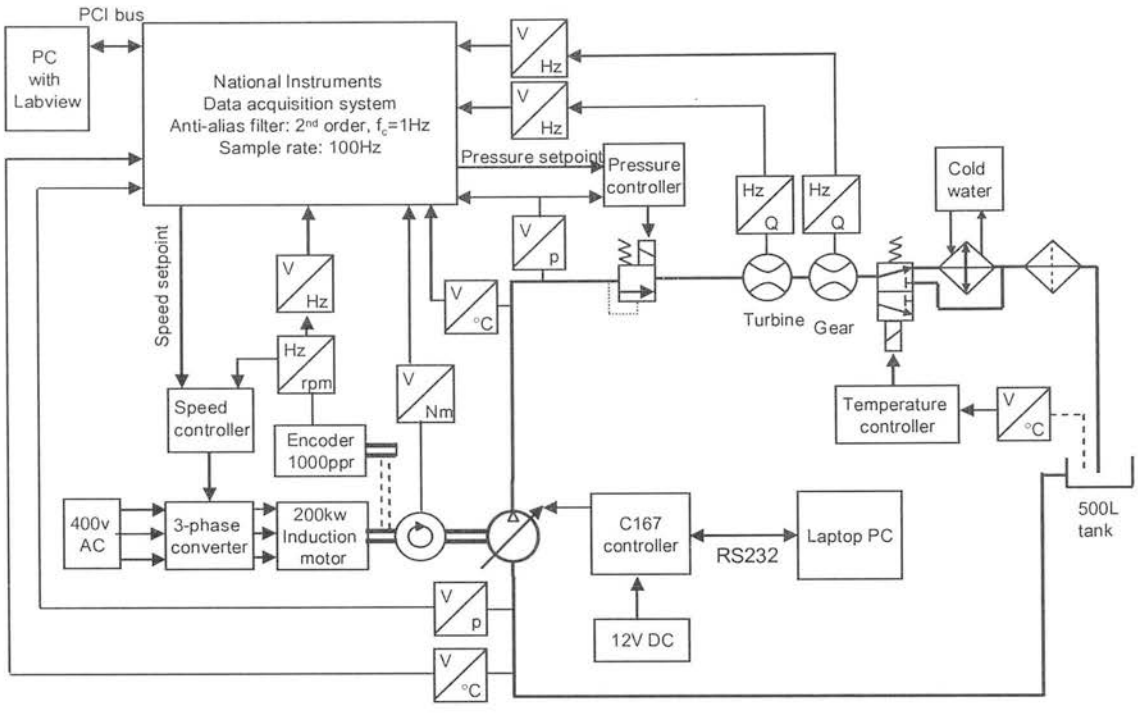


Figure 2-9: Test rig for the SD1B tests

The tests were conducted at the Sauer Danfoss test facility in Ames (USA) by the author in conjunction with Luke Wadesley and Onno Kuttler of Sauer Danfoss.

The schematic of the test rig and data acquisition arrangment (DAQ) is shown above. Some of the important features are listed below:

- There was closed-loop control of prime-mover speed, pump output pressure and tank temperature.
- The speed and pressure setpoints were generated by the DAQ, allowing the PC software to create control ramps of speed and pressure. For all of the efficiency tests, the speed was ramped from 500 to 3200rpm in 180 seconds; this was slow enough such that the torque needed to accelerate the inertia of the pump was negligible. Using a slow speed ramp rather than discrete speed points minimised the number of individual tests needed to cover the full operating range.
- Two flowmeters were fitted, one of the positive displacement gear type, the other a turbine type. This was to investigate the effect of the DDP pulsation on the quality of the flow signal from each type of transducer. Five metres of low-pressure hose connected the PRV outlet to the flowmeters, which in conjunction with the significant pressure drop of the gear flowmeter created a low-pass filter; this was important as

neither unit had quadrature pickups and so any momentary backwards flow would generate measurement error. No problems with pulsation were detected during initial tests so the signals from the two flowmeters were averaged for all subsequent measurements; they agreed within 0.5%.

- As the flow was measured at low pressure, downstream of the pressure relief valve, the measured flow was slightly higher than the actual flow from the pump at high pressure, due to the bulk modulus and the thermal expansivity of the fluid. This effect was corrected by scaling down the flow measurements as per Section 3.1.3 of ANSI/NFPA T3.9.17, and all subsequent analysis was based on these scaled values; details of this calculation are shown in Appendix 9.1.

2.5.2 SD1B test plan

A test plan was drawn up with the aim of spanning the complete range of pressure, speed and fraction of full displacement (F_d). Each test was given a code number; these are tabulated below:

Pressure (bar) F_d	<5	50	100	150	200	250	300
0%	100						
12.5%		102	107	112	117	122	
25%		103	108	113	118	123	
50%		104	109	114	119	124	
75%		105	110	115	120	125	
100%	101	106	111	116	121	126	127

Figure 2-10: SD1B test plan

Individual runs were also made for special purposes; these are mentioned in the appropriate section below.

All of the measured quantities were measured by a 16-bit analogue to digital converter at 100Hz sample rate. A 2nd order (40db/decade) filter with cut-off frequency of 1Hz ensured that aliasing was all but eliminated, providing 67dB of attenuation at the Nyquist frequency of 50Hz. Sampling 11 channels at 100Hz for 180 seconds yielded 198,000 measurements for each run. To minimise this for subsequent analysis, each run was processed through a re-sampling algorithm, outputting the values at intervals of 100rpm from 500 to 3200rpm. This was done by averaging the measured values over -50 to +50 rpm of the interval, reducing the number of data points per run to a more manageable 297.

All the tests described herein were conducted with fluid temperature of 50°C (+/-1°C).

2.5.3 SD1B results

Raw data from one speed sweep

Below are the results from test 121, a sweep of speed at 200 bar and at 100% displacement.

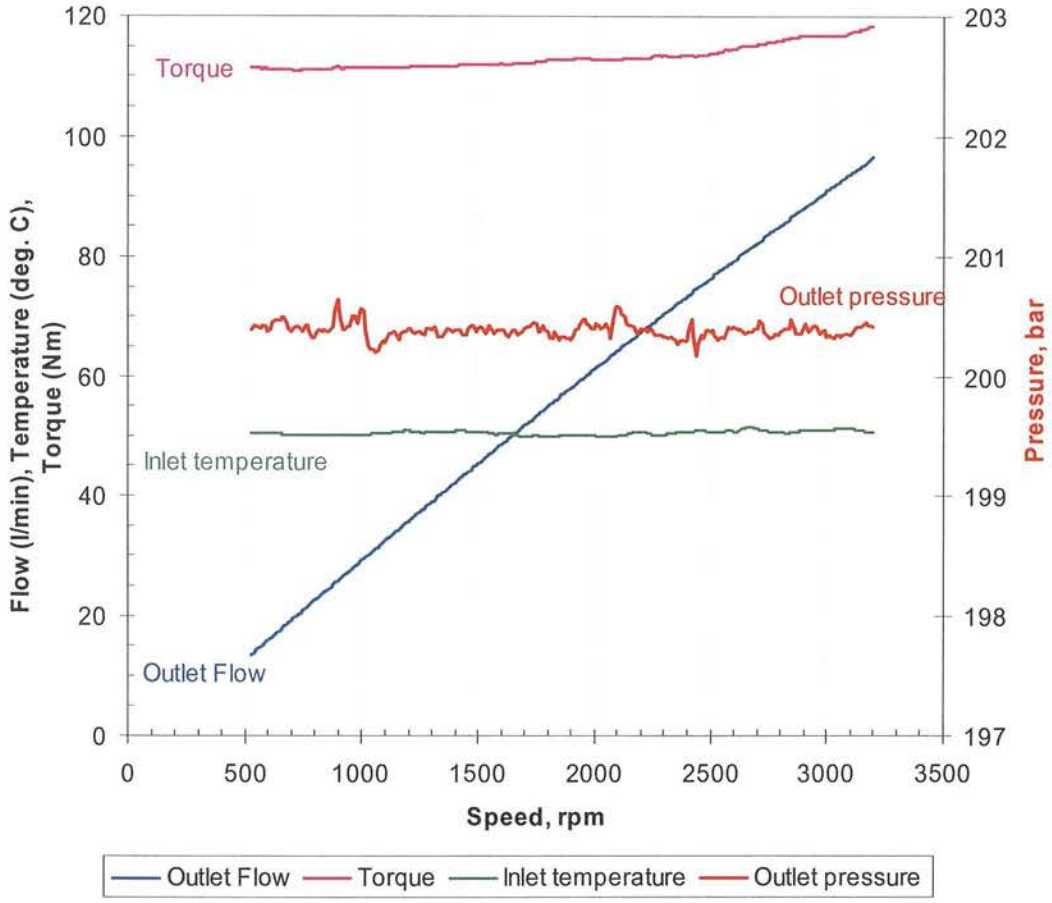


Figure 2-11: SD1B; Results from test 121 (filtered signals from transducers)

There are some notable features in the above figure:

- The outlet pressure is kept very constant (+/- 0.2 bar) by the closed-loop electronic pressure controller, despite the large increase in flow from the start to the end of the test.
- The inlet temperature is well controlled to 50°C +/-1°C.
- The outlet flow is fairly linear with respect to speed, and the cylinder filling was sufficient to allow operation at 3200rpm with an atmospheric intake.
- The input torque rises with speed, an indication of speed-related torque loss.

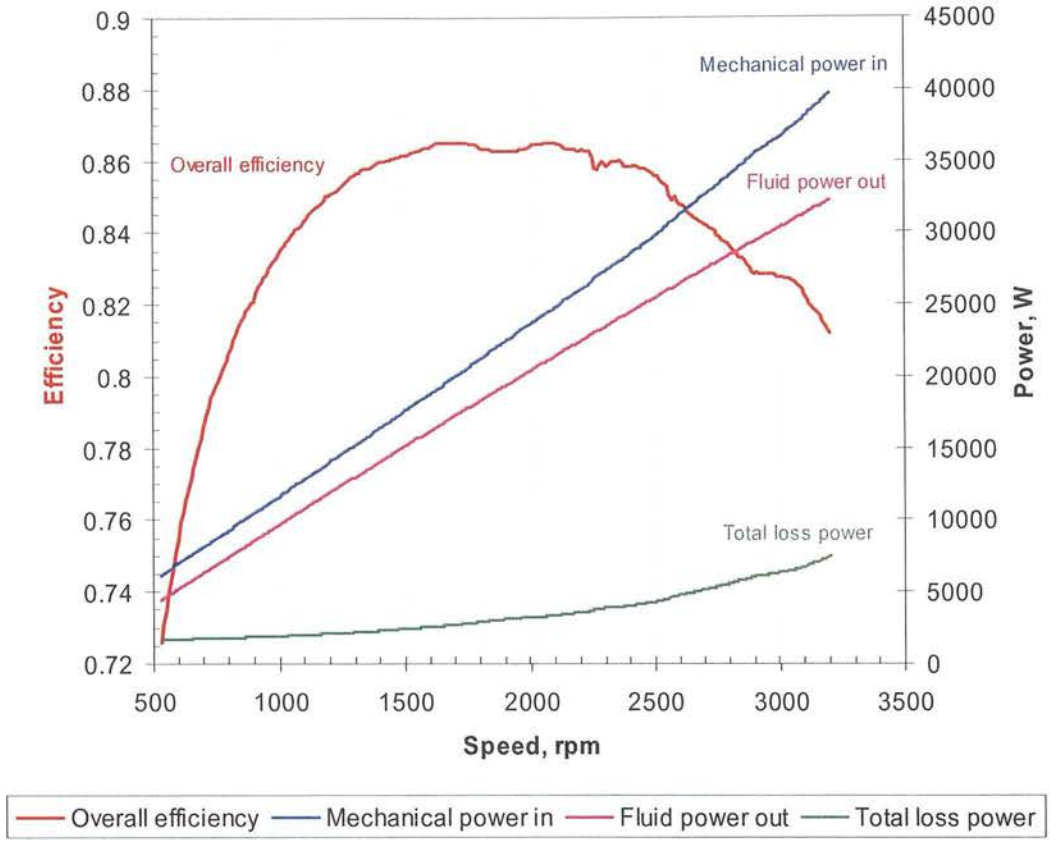


Figure 2-12: SD1B; basic analysis from test 121

For this test the overall efficiency peaked at 86.5%, with pronounced roll-off at low and high speed, as is typical of a positive displacement pump.

Effective displacement with speed and pressure

Below are the output flow results of all the SD1B tests with a full displacement demand (test nos. 101, 106, 111, 116, 121, 126, 127).

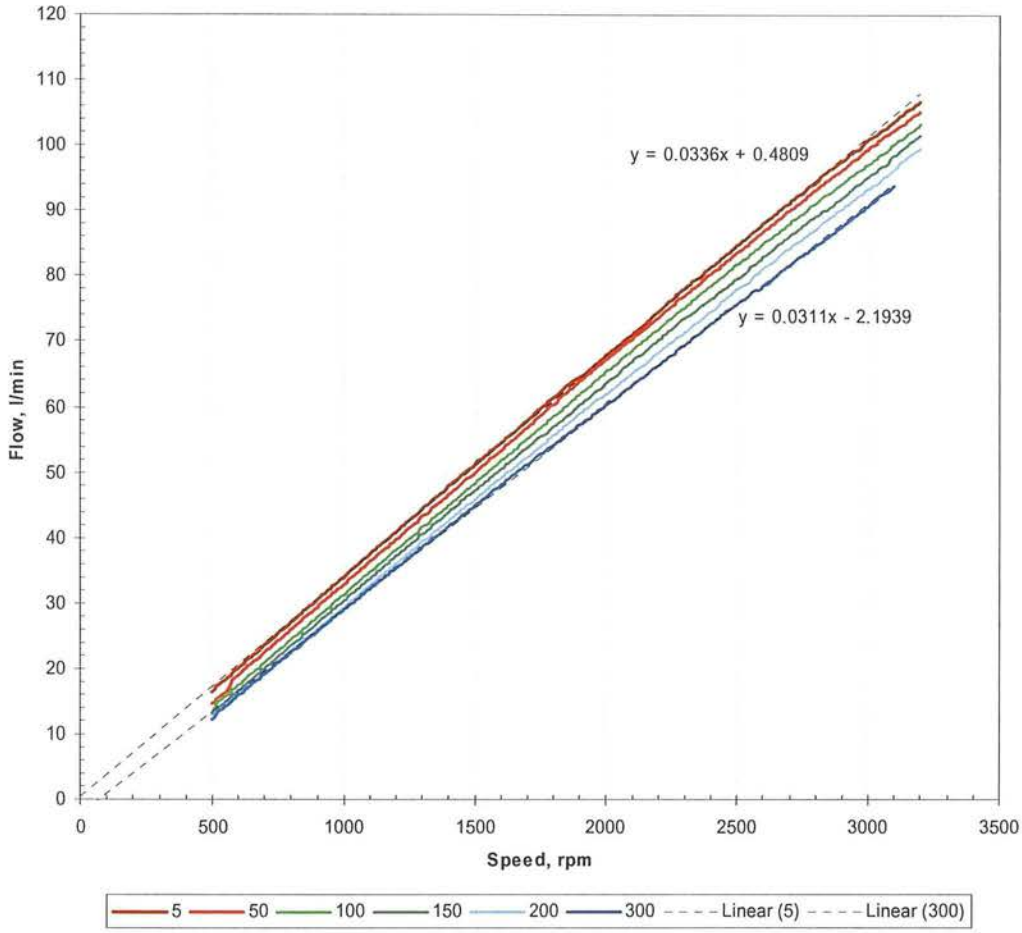


Figure 2-13: SD1B; Flow vs. speed, lines of pressure (bar)

The outlet flow shows a reasonable linearity with respect to speed. The effect of increasing pressure both reduces the gradient of the line and introduces a negative offset; these are effects expected because of compressibility and leakage respectively.

Dividing flow by speed shows the effective displacement for these tests (V_{eff}), which is shown below:

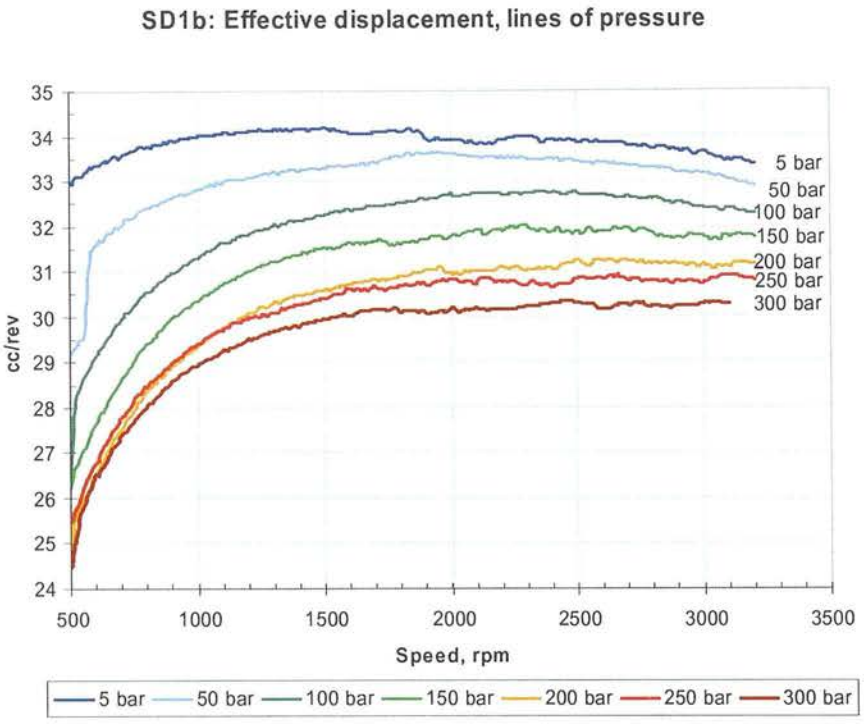


Figure 2-14: SD1B: V_{eff} vs. speed; lines of pressure

In the figure above, the expected manifestation of leakage and compressibility can be seen. At high speed, leakage is not so significant, compressibility effects dominate and the line of V_{eff} become horizontal with an increasing downwards offset as a function of pressure. As speed reduces, the lines curve downwards due to the increasing relative effect of leakage compared to output flow. Note that the inflections on the 50 and 100 bar lines below 600rpm are thought to be due to the hydraulic circuit not having reached a steady state before the start of the speed ramp.

The shape of the 5 bar curve in Figure 2-14 deserves more explanation. The displacement of the machine was observed to change as a function of speed, at a pressure where the leakage and compressibility effect should be minimal. There are a number of possible explanations of this:

Inlet cavitation. If the pump was suffering from inlet cavitation, this would be expected to cause a dramatic collapse in flow above a critical "fill speed", not the observed gentle roll-off; this effect is discounted.

Timing inaccuracy of the LPV. Because the DDP is commutated by the controller sending pulses to the LPV, the phase accuracy of the commutation is only as good as the timing

accuracy of the controller and the consistency of the actuation delay of the LPV. These were carefully calibrated before testing but errors may have persisted.

Dynamics of the passive HPV. The HPV in the SD1B is a spring-return poppet valve which has its own dynamic behaviour (similar to the delivery valve in Edge et al., 1990). If the HPV is not fully closed by the end of the pumping stroke, it is possible for some fluid to flow back into the cylinder after it passes TDC, possibly causing a reduction of the effective displacement of the pump. This effect was investigated at the earlier calibration stage and was found to be clearly significant at speeds above 2000 rpm.

Pressure peaks in the working chamber. Although the outlet pressure was only 5 bar, the actual cylinder pressure may have been higher due to pressure drops caused by the impedance of the HPV and the high-pressure flow galleries. Clear evidence for this can be seen in the difference between losses when pumping at full displacement at low pressure, and the idle losses, as shown in Figure 2-16 below.

These results are presented as a function of pressure below:

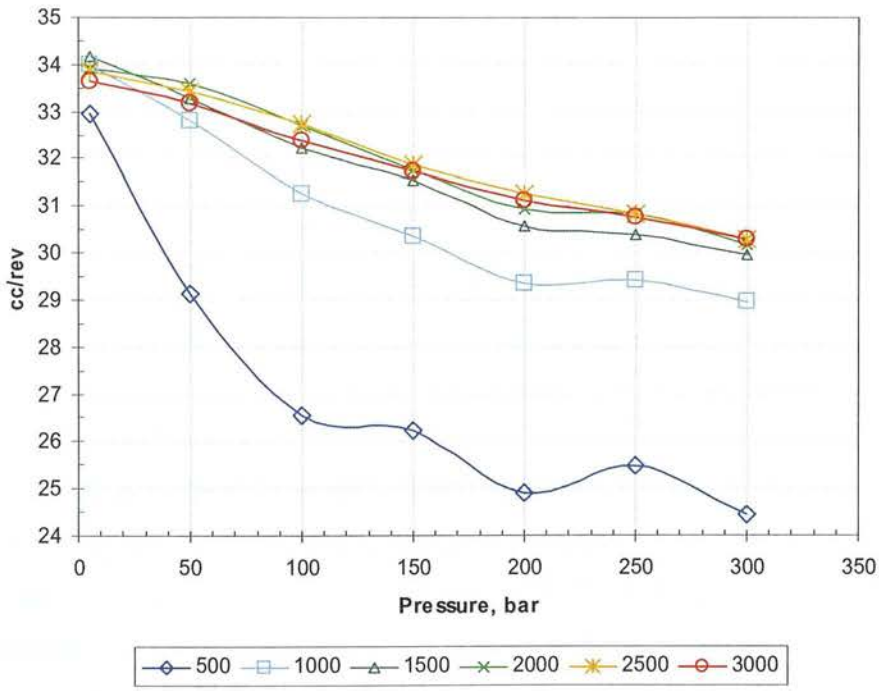


Figure 2-15: SD1B: V_{eff} vs. pressure; lines of speed

The above figure shows a clear linear reduction of V_{eff} with respect to pressure at the higher speeds; this is the expected effect of the reduction in displacement due to fluid compressibility. Below 1500rpm, leakage flow has a more significant effect on V_{eff}, because at lower speed the output flow is lower.

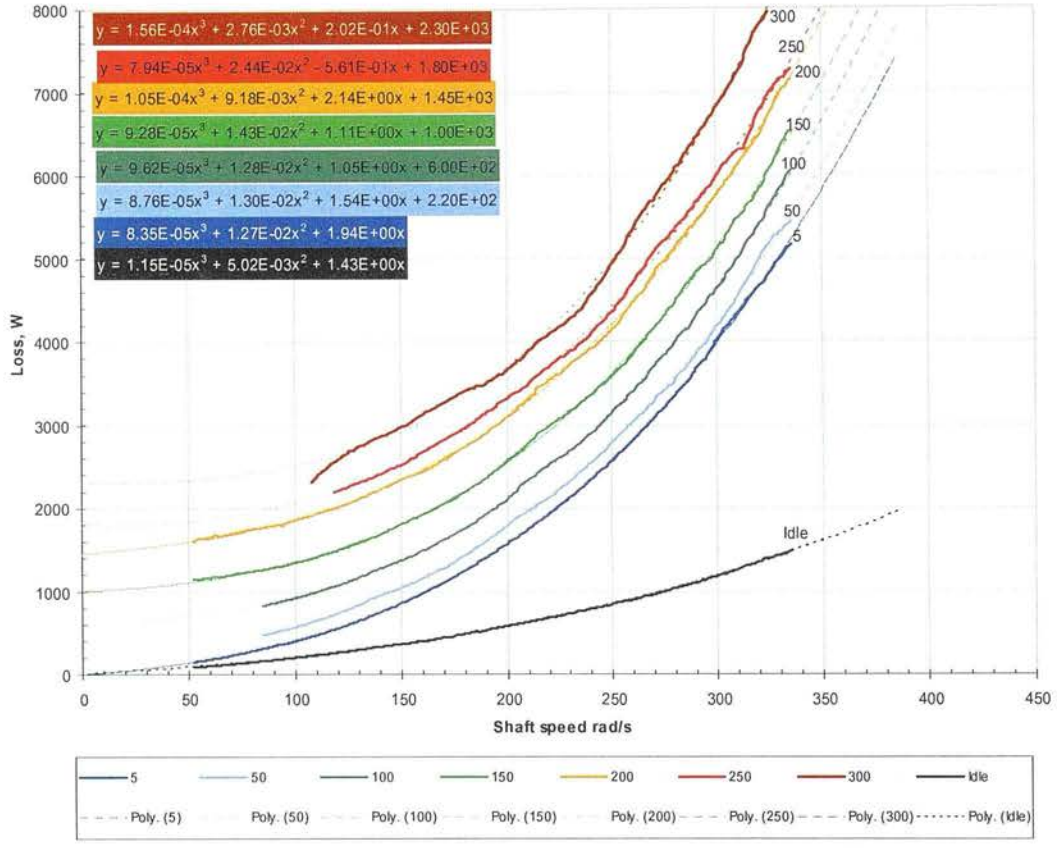


Figure 2-16: SD1B; Total losses as a function of speed (rad/s); lines of pressure

The figure above shows the total losses P_{loss} at full displacement, as a function of speed, at a selection of pressures. Also included is the idle loss curve, when output flow was zero. Each curve is fitted with a 3rd order polynomial by the least squares method (shown as dotted lines), which would appear to provide a reasonable fit for all the curves.

When the machine is idle or is delivering flow at very low pressure, it is to be expected that there should be zero loss at zero speed, and indeed a good fit of the idle and 5 bar lines was obtained with no zero-order coefficient.

There is obviously a very large difference between the idle losses and the losses when pumping at 5 bar. The SD1B appeared to have very substantial internal pressure drops. It can be seen that at 3000rpm and 300bar, the flow-related loss is greater than the pressure-related loss. The explanation for this is that the SD1B machine retained many flow passage dimensions from an earlier machine designed to operate at 1500rpm, and this has caused a substantial compromise in its energy efficiency at high speed.

As pressure increases, the loss curves show a clear positive offset but have a similar shape to the 5 bar curve. The losses are plotted against pressure in the figure below.

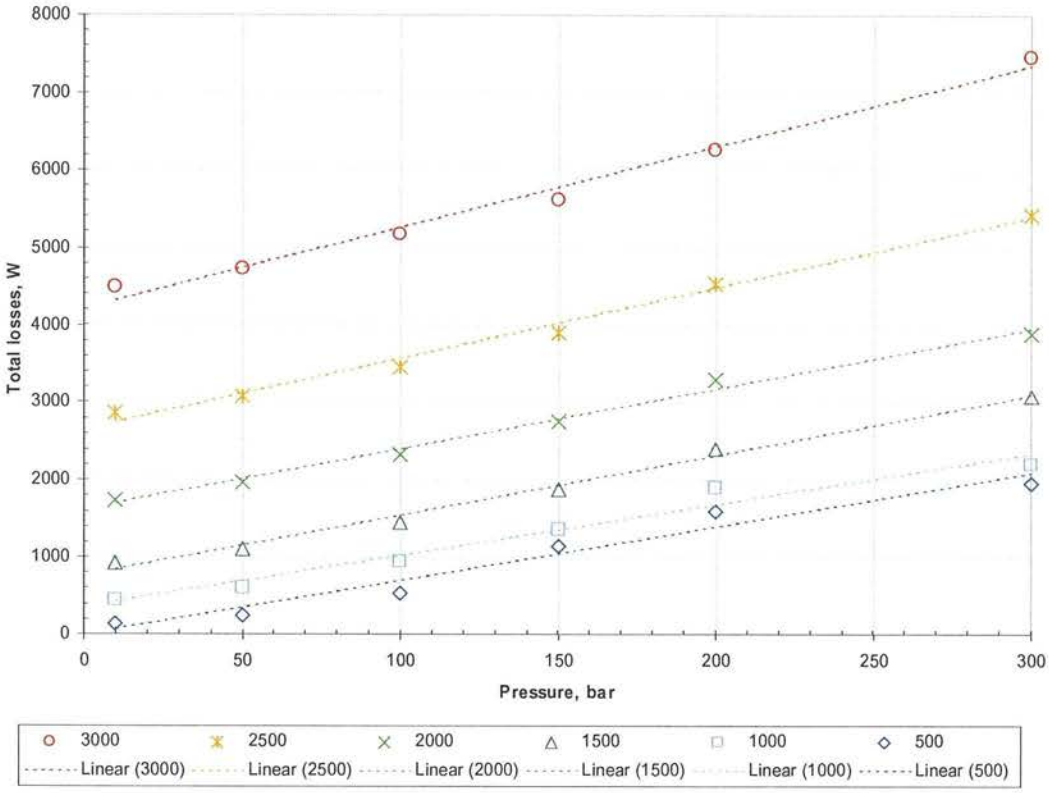


Figure 2-17: SD1B; Total losses as a function of pressure; lines of speed (rpm)

The figure above shows that total losses were reasonably linear with respect to pressure. Due to this result, an assumption of linearity was used in the later mathematical model.

Analysis of the idle losses

One component of the idle loss that it is relatively easy to separate from the others is the loss due to the pressure drop in the disabled (open) LPVs. When the DDP is at idle, the flow into and out of the pumping chamber is impeded by the disabled LPV. The characteristics of this valve were well known from separate tests by Fergus Macintyre of AIP on a specially-built reciprocating flow test rig.

In these tests it was found that, with respect to flow rate, the pressure drop across the disabled LPV has a first order (viscous shear) and a second order (kinetic energy) component. These first and second-order coefficients were applied to the sinusoidal flow-rate, calculated from the displacement and speed of the cylinder, to predict the instantaneous loss power due to the pressure drop across the LPV, as shown below.

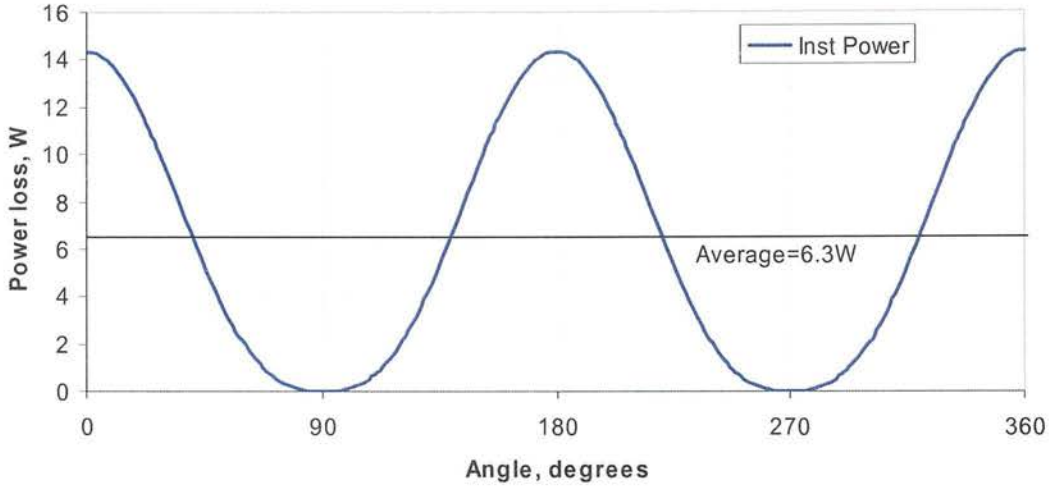


Figure 2-18: SD1B; Calculated instantaneous idle losses per cylinder due to LPV breathing at 2000rpm.

The predicted average power due to LPV breathing at 2000rpm is 6.3W per cylinder.

This calculation was repeated across the full speed range, multiplied by 18 cylinders, converted to a torque loss and graphed as a function of speed to calculate an estimate of the idle torque losses attributable to LPV breathing.

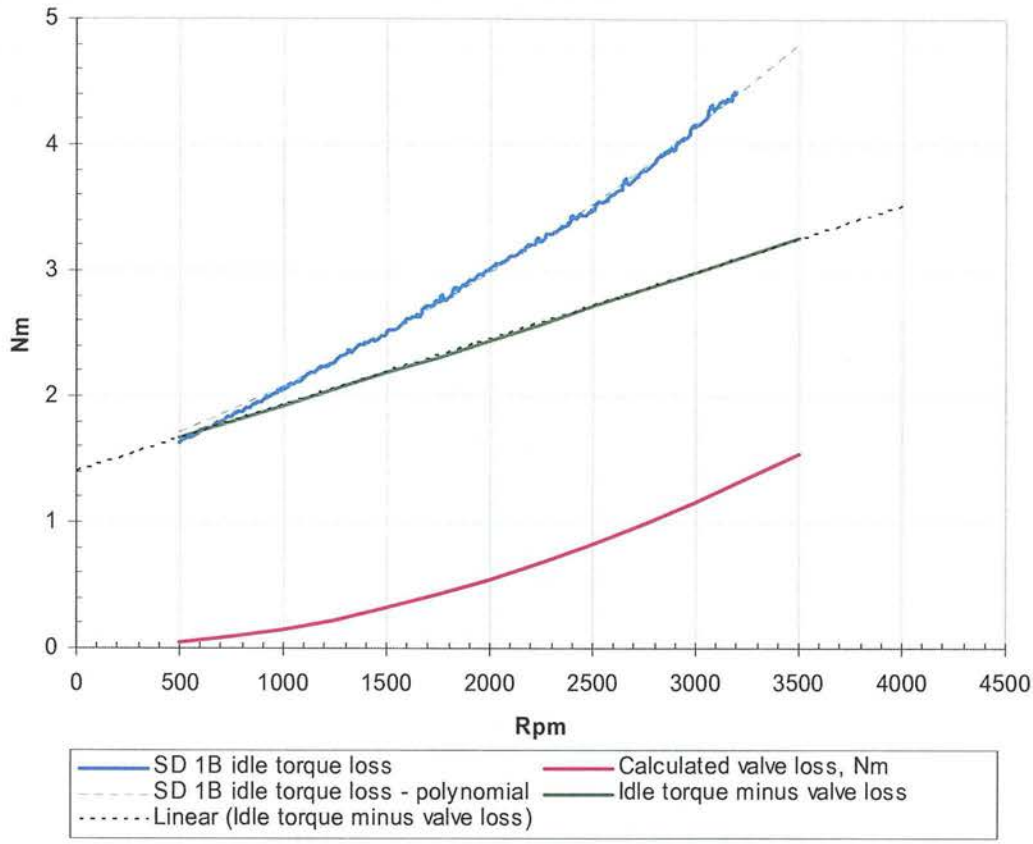


Figure 2-19: SD1B; idle torque loss analysis

The blue line above shows the measured idle torque loss (Nm) with a 2nd order polynomial fit (dotted line). The pink line above shows the torque loss which was calculated to be due to LPV breathing losses. The green line shows the result of subtracting the calculated valve torque loss from the measured torque loss. The remaining torque loss is a straight line with a positive offset at zero speed.

This analysis shows that the second-order component of idle torque loss can be completely attributed to the LPV breathing losses. The constant component is consistent with the Coulomb friction from the spring force keeping the piston pad in contact with the crankshaft, whilst the linear component is consistent with viscous shear between piston and cylinder, and within the rolling bearings.

Splitting losses into torque and flow losses

As noted in section 2.6 the splitting of losses into 'volumetric' and 'mechanical' requires knowledge of k_{dead} for the particular machine under test. With access to the CAD files used to design it, this is relatively straightforward.

To calculate the total pressurised volume in each chamber, a special part was created in the solid model assembly which was generated by subtractive geometry from the surrounding solid parts. This produced a solid part corresponding to the interior pressurised space of the pumping chamber at TDC, analogous to pouring wax into the pump to take a cast of the interior space. The volume of this part can be calculated automatically by the Solidworks 2003 software.

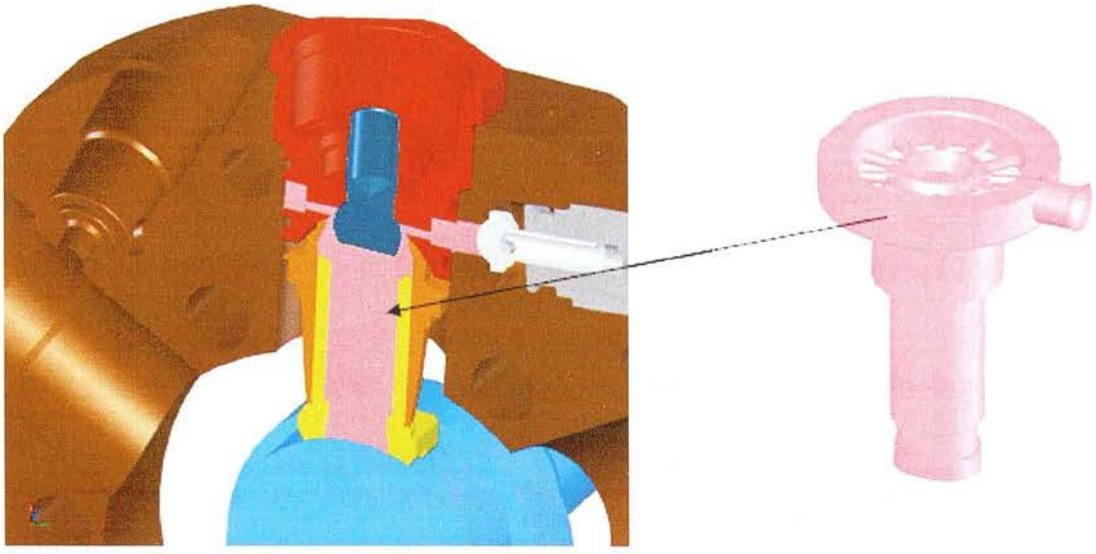


Figure 2-20: Calculating the pressurised volume of the SD1B with Solidworks 2003

The results of this calculation were:

Swept volume of one pumping chamber= $V_{geom}/18 = 1.94$cc

Dead volume of one pumping chamber = $V_{dead}/18 = 5.43$cc

$$\Rightarrow k_{dead} = 2.80$$

Using this figure for k_{dead} allows the calculation of V_{comp} as a function of pressure as per eq. 18 (p.21), while eq. 15 (p.20) allows the calculation of V_{leak}, and therefore Q_{leak}. Thus the leakage flow can be derived from the experimental results by first making an estimate of the flow loss due to the compressibility effect.

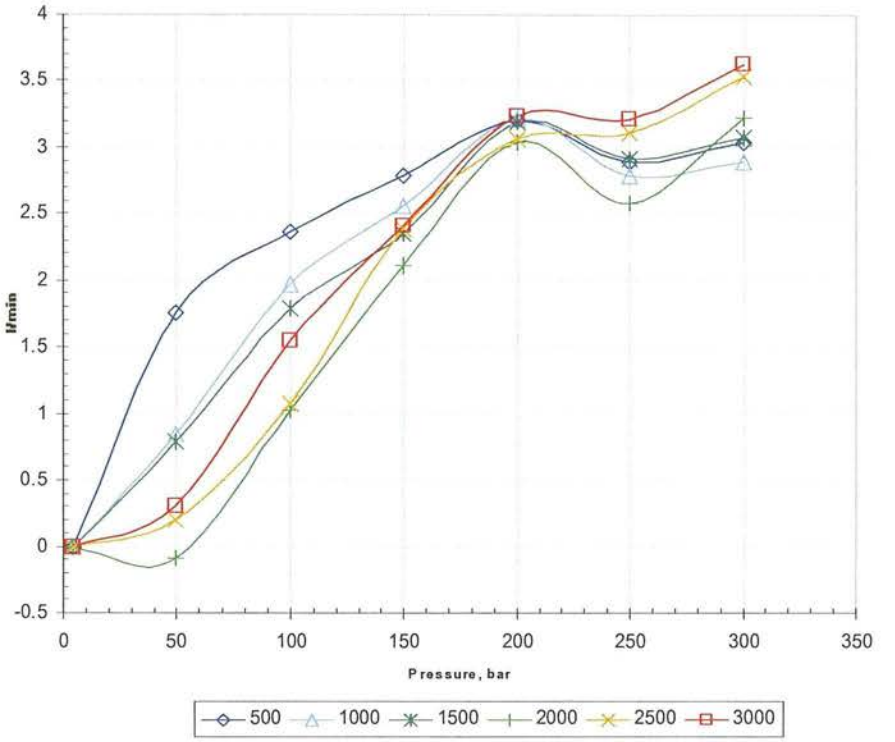


Figure 2-21: SD1B; Derived leakage flow (l/min) as a function of pressure (bar); lines of speed (rpm)

The figure above shows the leakage flow derived as described in the previous paragraph. It is notable that the leakage seems to level off at higher pressure; this is consistent with mechanical strain causing the leaking gaps to reduce as pressure rises.

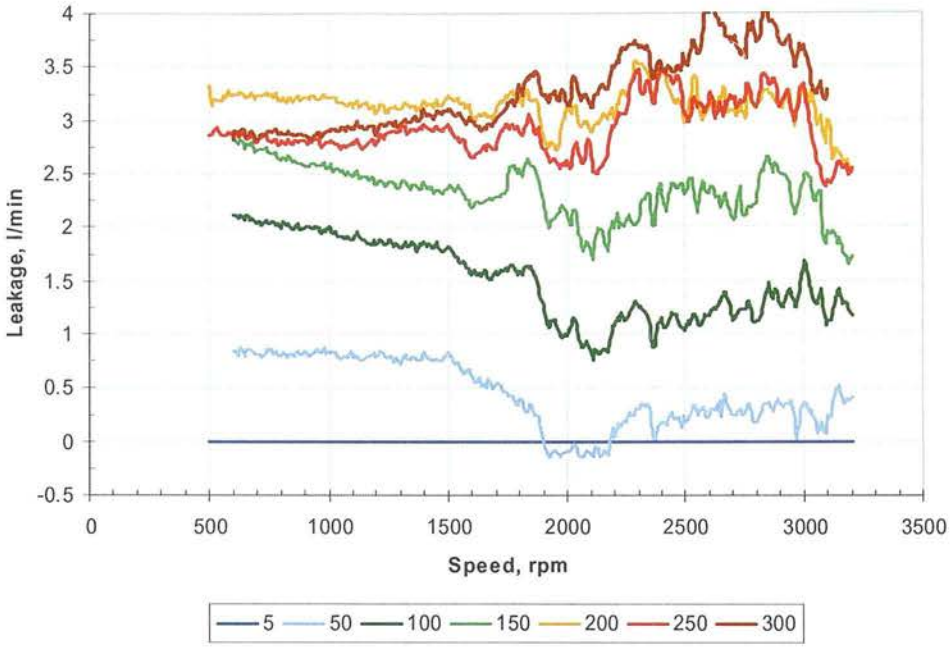


Figure 2-22: SD1B; Derived leakage flow as a function of speed; lines of pressure

The above figure shows the leakage flow as a function of speed at a range of pressures. The behaviour here is quite complex, but as speed increases there is a clear downward trend at low pressure, whilst at high pressure the leakage is more constant.

With the calculation of the leakage loss Q_{leak}, the torque loss T_{loss} can be calculated from eq.18 (page 21). This is termed the 'derived torque loss'.

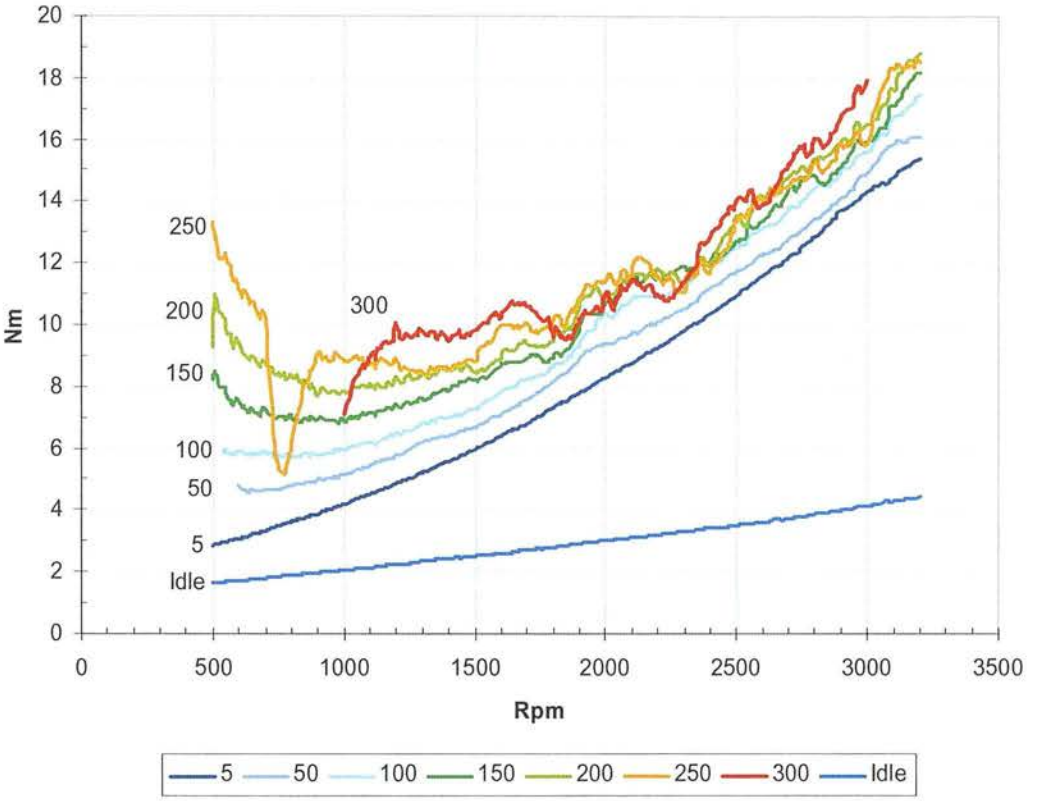


Figure 2-23: SD1B; Derived torque loss as a function of speed (rpm); lines of pressure (bar)

The figure above shows a pronounced increase in torque loss at high speed which is common to all pressures; this can be attributed to the flow-related pressure drops inside the pump. There is also a substantial increase in torque loss at low speed; this is likely due to reduced hydrodynamic lubrication resulting in higher bearing friction, most likely in the piston pads. Similar behaviour has been reported for a gear pump (Dorey 1988).

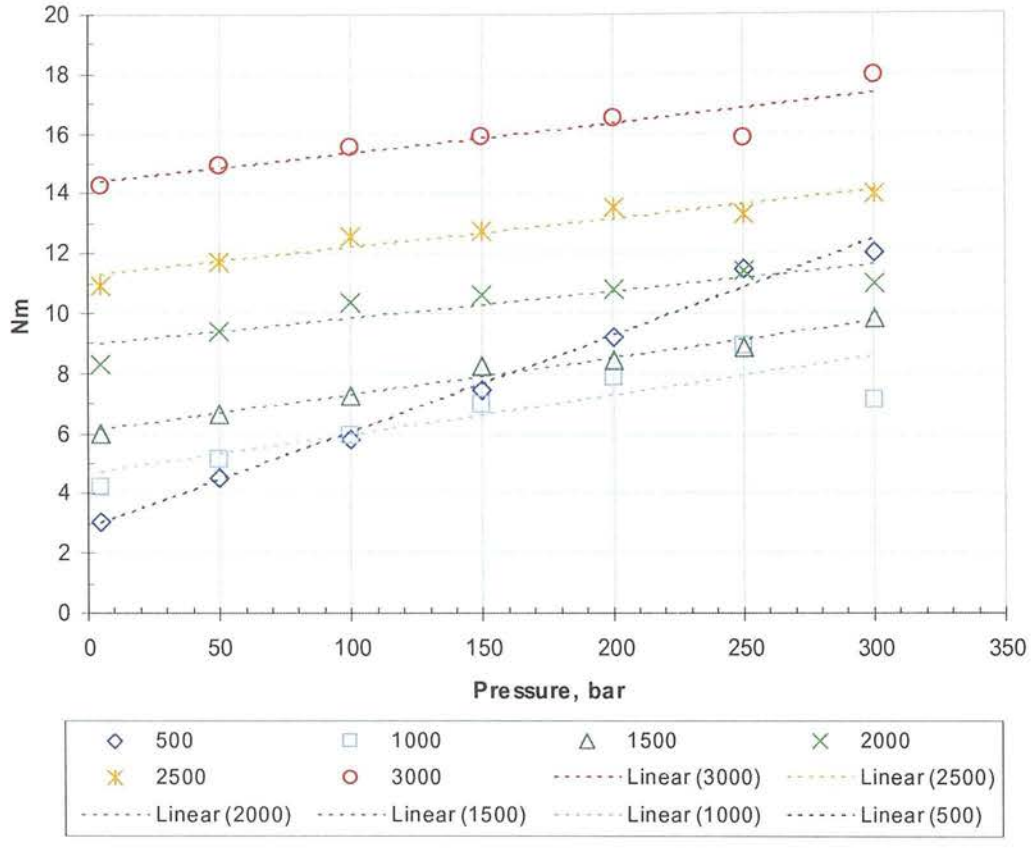


Figure 2-24: SD1B; Derived torque loss as a function of pressure; lines of speed (rpm)

The derived torque losses show a clear linear relationship with pressure.

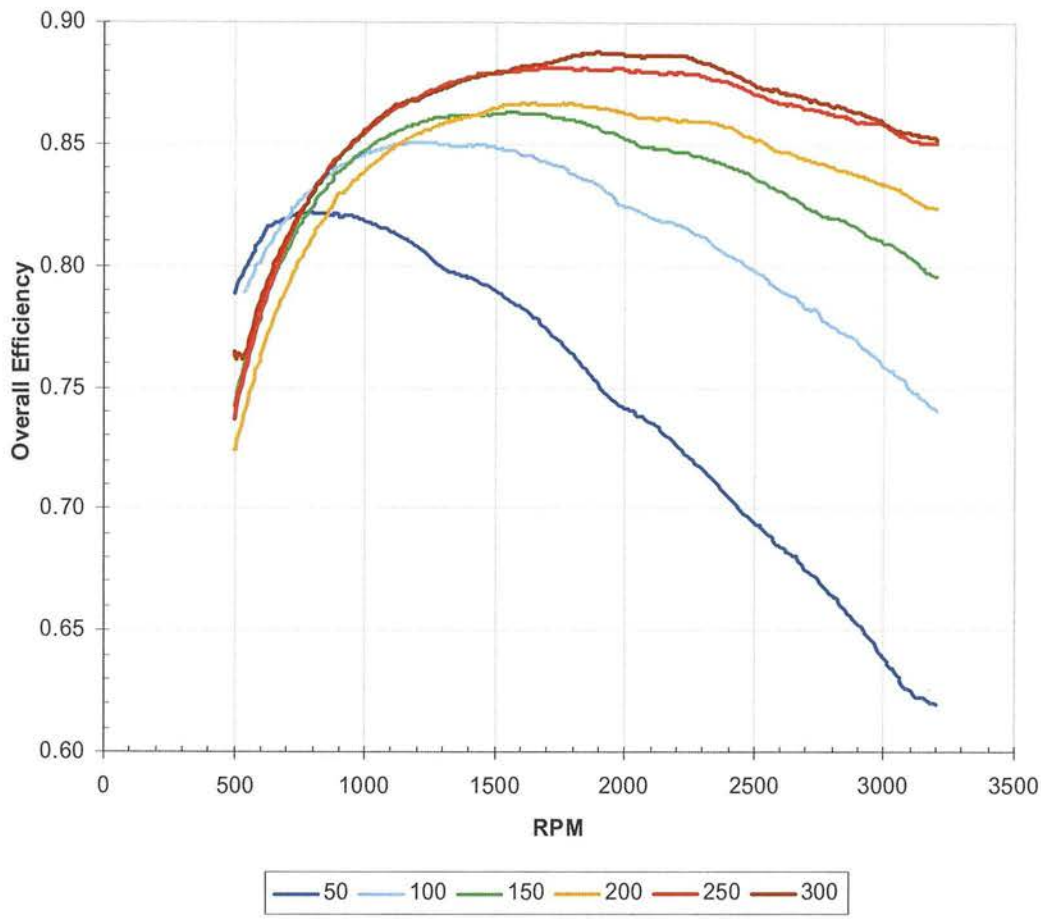


Figure 2-25: SD1B; Overall efficiency as a function of speed; lines of pressure (bar)

The overall efficiency peaked at 88.5% at 300 bar, 2000rpm. At all pressures, the efficiency can be seen to roll-off at high and low speed. The speed at which peak efficiency occurs can be seen to increases with pressure; at low pressure, the internal flow losses become more significant and leakage less significant, and therefore the peak occurs at low speed, whilst this situation reverses at high pressure.

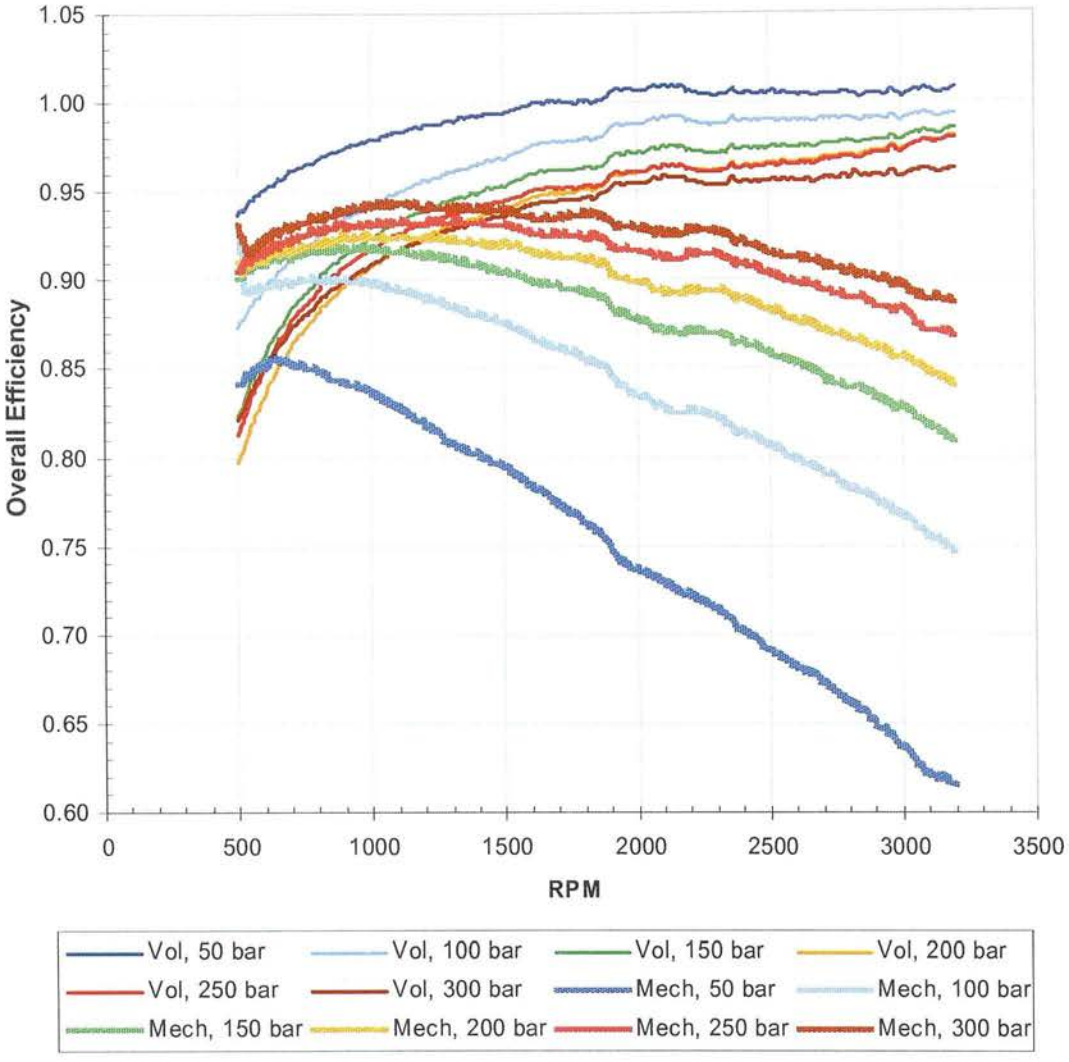


Figure 2-26: SD1B; Volumetric and mechanical efficiency by revised definitions as a function of speed; lines of pressure

The figure above shows the volumetric and mechanical efficiency according to the revised definitions in Section 2.4.

The volumetric efficiency can be seen to fall at lower speed, where leakage is more significant, reaching a peak at maximum speed.

The mechanical efficiency can be seen to drop at higher speed, due to the higher internal flow losses; and at lower speed, due to reduced hydrodynamic lubrication effect.

Linearity of total losses with respect to displacement fraction

For formulating the loss model, it is important to investigate the effect of varying the displacement fraction on the total losses.

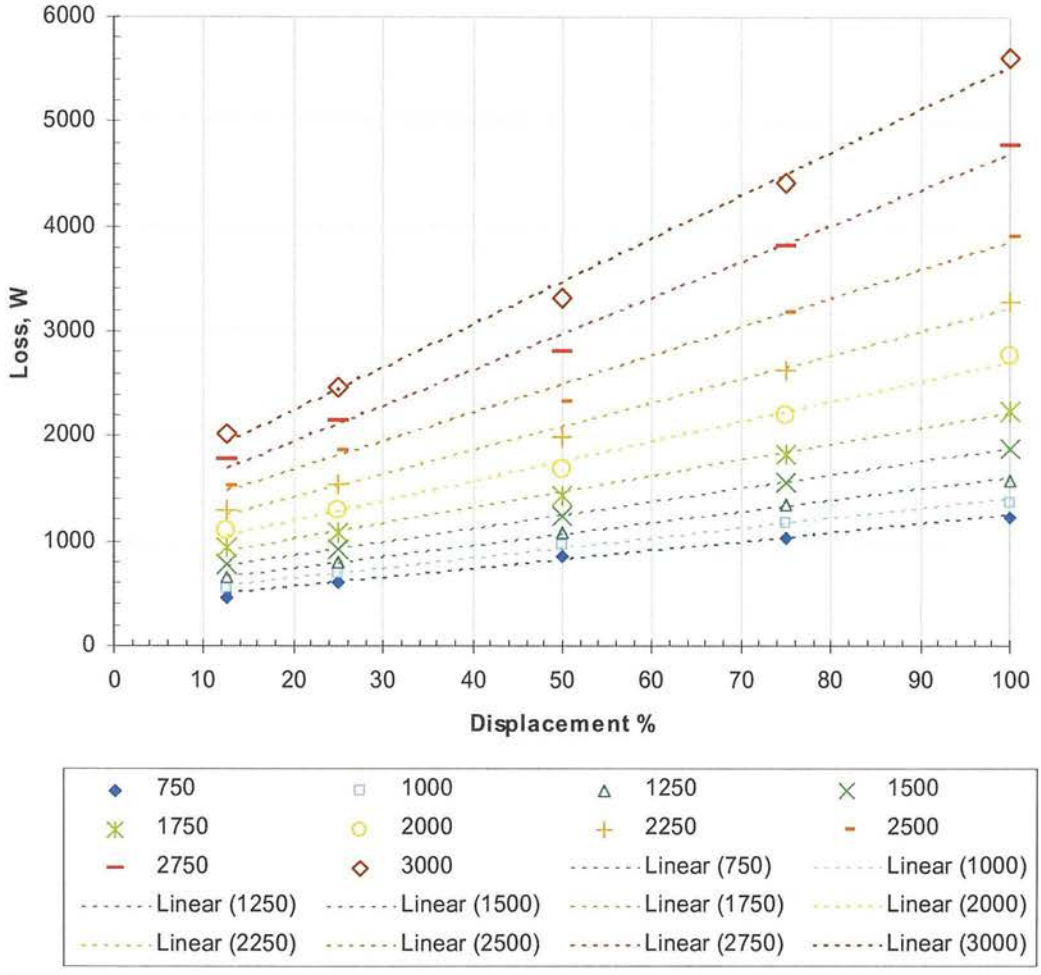


Figure 2-27: SD1B; Variation of P_{loss} as a function of F_d; lines of rpm

From the limited spread of displacements used in the efficiency tests, the losses appear to be linear with respect to F_d, which is the behaviour expected due to the enabling fraction method of displacement control.

To verify this with a continuous spread of F_d, a special experiment was carried out to sweep the DDP up and down through a continuous range displacement demand, using the analogue voltage demand from the DAQ (where 0V was zero displacement, and 4.5V was full displacement). This also allowed flow control linearity and hysteresis to be inspected.

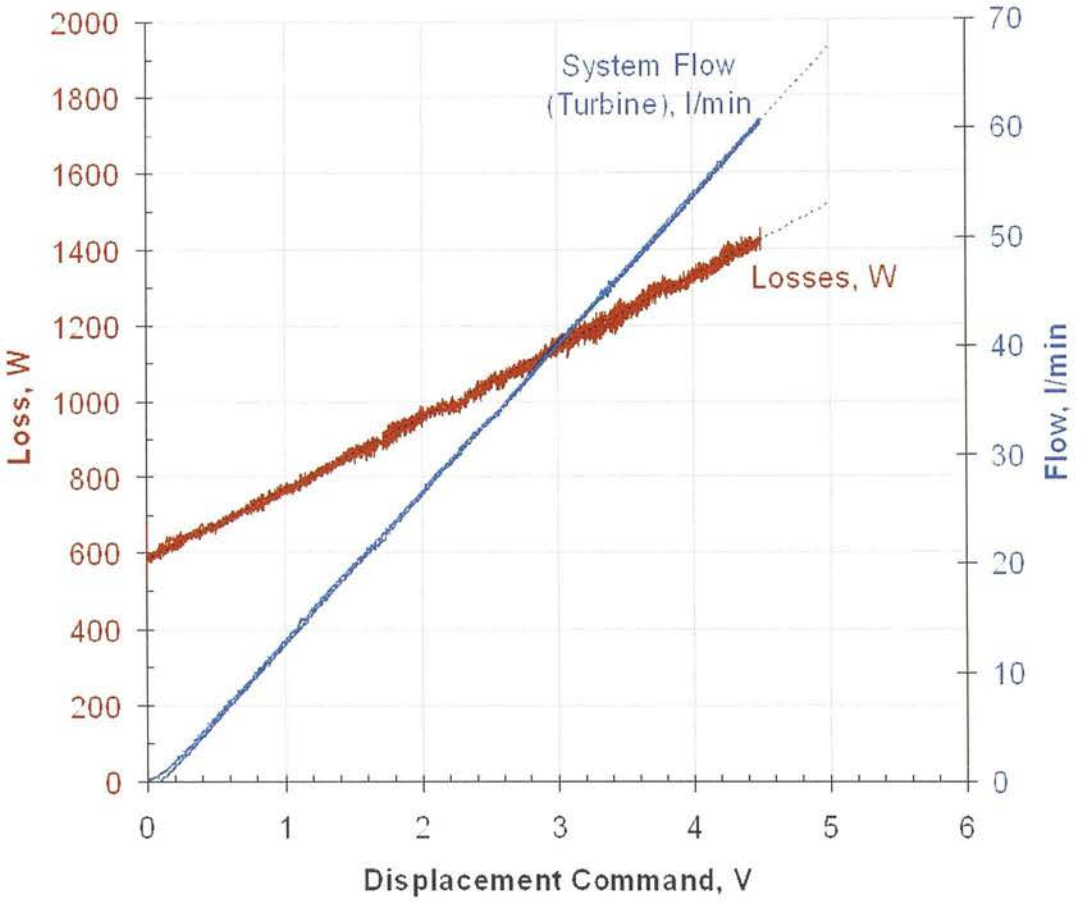


Figure 2-28: SD1B; variation of Q and P_{loss} as a function of F_d; 1800rpm, 5 bar load

The losses can be clearly seen to be linearly proportional to displacement fraction, confirming the results of Figure 2-27.

The flow from the DDP can be seen to be linear with respect to commanded displacement; the "dead band" which is typical of variable-stroke pumps using an electrohydraulic actuator is not present. This is to be expected because the generation of variable flow is done by cylinder enabling. There is no measurable hysteresis - this is to be expected because the pump essentially has no mechanical memory of what has gone before; there is no swashplate mechanism with associated friction and damping.

It has been clearly shown that the total losses in a DDP are linear with respect to displacement fraction. This confirms the basic theory of cylinder disabling as a displacement control mechanism, and greatly simplifies the task of modelling the losses.

However plotting losses vs. F_d at one speed and a range of pressures revealed some non-ideal behaviour, as shown below:

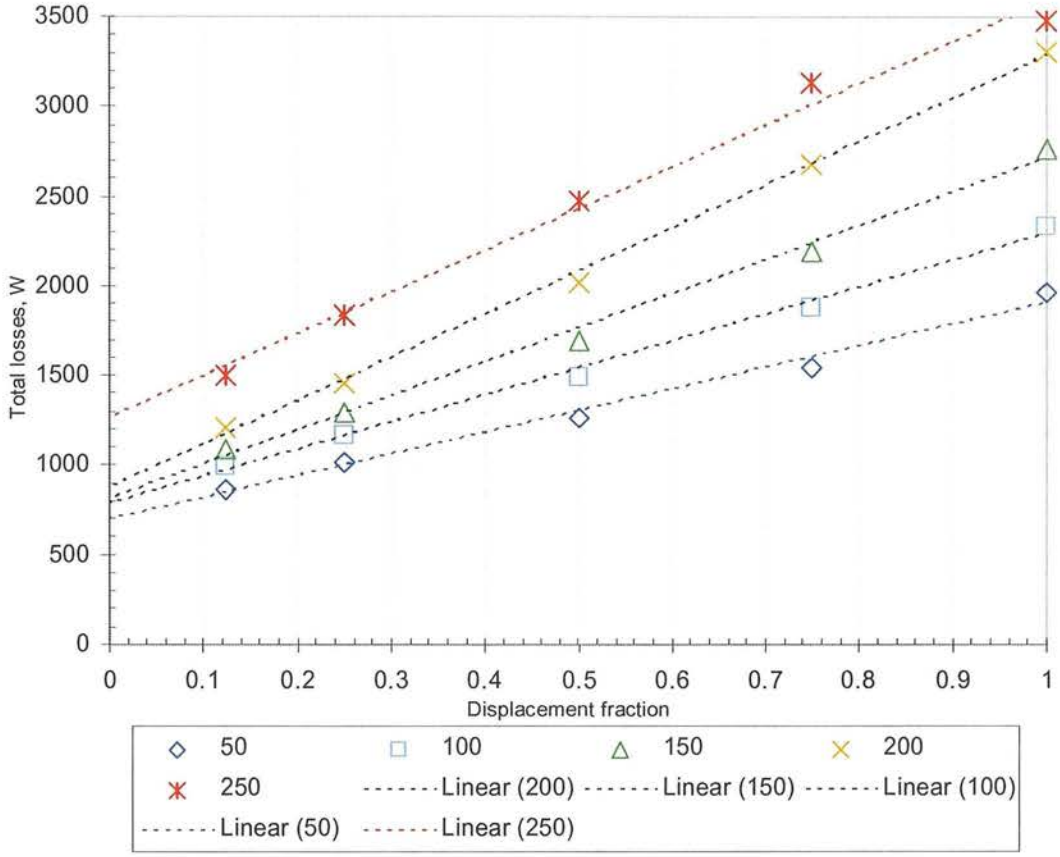


Figure 2-29: SD1B; variation of P_{loss} as a function of F_d at a range of pressures; 2000rpm

Theoretically, all of the graph lines lines should intersect the F_d=0 axis at the same spot; the losses of the idling DDP should be the same at all pressures. Leakage at F_d=0 should be zero, because all of the leakage paths are upstream of the HPVs, which at F_d=0 should all be closed and sealing perfectly (tests on the HPV in isolation indicated that it leaked at only a few drops per minute at max pressure). This indicated that either torque or leakage losses were not behaving as expected at high pressure, although the effect was small.

Plotting input torque vs. F_d investigated whether torque losses are therefore implicated:

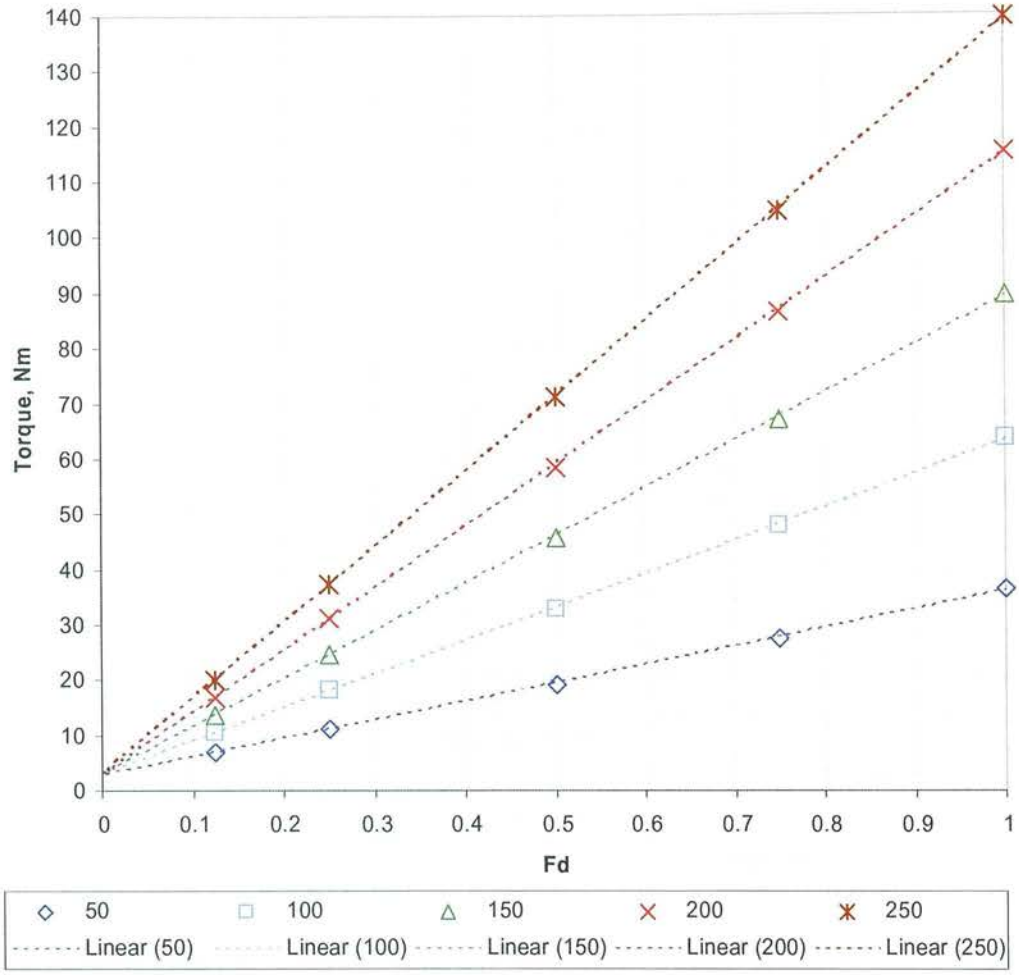


Figure 2-30: SD1B; variation of input torque as a function of F_d at a range of pressures (bar); 2000rpm

The above figure shows clear linearity, with all pressure lines intersecting at one point, which is the idle loss for this speed. This indicated that the problem lay with leakage not reducing to zero at zero displacement fraction, because there was a leakage path in the pump upstream of the HPVs.

Considering Figure 2-29, at 250 bar, zero displacement, 2000rpm there was an extra 600W of loss compared to the losses at zero pressure, corresponding to a parasitic leakage of 1.4 l/min.

When the tests were completed an explanation was found for this non-ideal leakage behaviour.

Figure 2-31: Extrusion of seals in the SD1B, causing parasitic leakage

The photo above shows the view of the pump with the pump-side endplate removed. The o-rings which were meant to seal the axial flow galleries had extruded at some point during the test. This is thought to be due to deflection of the pump-side end plate; this plate was not thick enough and was distorting under pressure. The area of the red seal visible above is under the full outlet pressure of the pump. The plate was clamped through the six holes at a smaller radius that the red seal, and the o-rings were at a greater radius. At low pressure the clamping load on the o-rings would be sufficient for the seals to work, but at some critical pressure the seals would leak. At this point the whole interface between pump ring and pump-side end plate would be forced apart by the pressure distribution of this diverging leakage from the axial galleries to tank pressure, at the outside of the pump ring. This would cause more plate deflection and a run-away situation would develop. This seal failure seems to explain the observed parasitic leakage.

A simple test to diagnose this failure was to pump the dead-headed DDP outlet to maximum pressure with a check-ball hand pump; this pressure was observed to be maintained for many hours if the seals were working correctly.

2.6 The proposed DDP loss model

The following objectives were identified for the loss model:

- The machine was assumed to operate at a steady state at a constant temperature.
- The model was to have the minimum number of parameters necessary, both for computational efficiency and to minimise the measurement effort needed to collect parameters. This meant taking advantage of as many simplifying assumptions as possible.
- It was to be formulated in such a way as to be useful for computer simulations of mobile machines.

Given the ambiguity inherent in splitting the measured losses into 'volumetric' and 'mechanical', the approach chosen was to model the total losses and split those into 'volumetric' and 'mechanical' at a later stage.

The model is formulated to account for the method of displacement control of disabling cylinders, rather than changing the piston stroke. The displacement of a DDP is varied by changing the fraction of cylinders which are enabled to those which are disabled. It is therefore reasonable to assume that the losses of a machine with varying displacement vary linearly with respect to displacement between the losses at zero displacement (i.e., at idle) and those at maximum displacement. This assumption appears to be valid from the SD1B test results presented in Figure 2-27 and Figure 2-28.

Furthermore, because the disabled cylinders are isolated from the pressure outlet, the idle losses (when all the cylinders are disabled) are assumed to be insensitive to outlet pressure.

2.6.1 Applying the DDP loss model to simulation studies

There are two main reasons to simulate the losses in a hydrostatic unit in a vehicle propel system:

- to calculate the overall fuel consumption and losses of the vehicle or mobile equipment;
- to investigate the effect of the losses on the control response and "drivability" of the vehicle.

These different objectives have led researchers to apply different formulations for the vehicle simulation model: backward-facing and forward-facing respectively. These types differ substantially in terms of strengths and weaknesses, simulation complexity and in requirements for the structure of the loss models for pumps and motors.

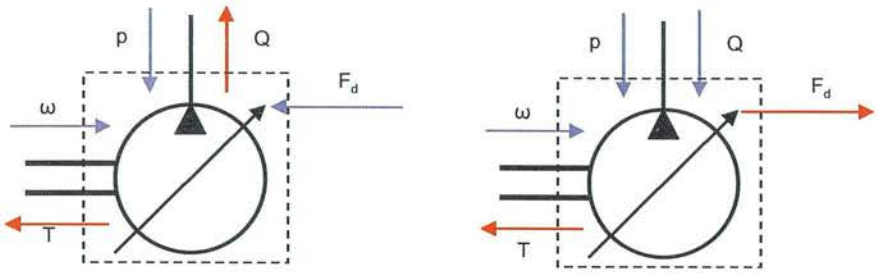


Figure 2-32: Forward-facing (left) and backward-facing (right) pump loss models

In the forward-facing formulation for a pump: speed, pressure and displacement fraction are inputs, while flow is an output.

In the backward-facing formulation for a pump: speed, pressure and flow are inputs, whilst displacement fraction required to achieve the specified flow is an output.

The DDP model presented here is forward-facing. In simulation studies of total losses over a drive cycle, it may be advantageous to convert this to a backward-facing formulation. Notes on how to do this are presented in Appendix 9.3.

2.6.2 Model of total losses

For a radial piston DDP, referring to the shape of the loss curves shown in Figure 2-16 (p.36), the power losses at zero displacement (idle) can be described by:

$$P_{idleloss}(\omega) = \begin{bmatrix} 1 & \omega & \omega^2 & \omega^3 \end{bmatrix} \cdot \begin{bmatrix} id_0 \\ id_1 \\ id_2 \\ id_3 \end{bmatrix} \tag{21}$$

In a DDP there can be no leakage back into a disabled cylinder if the HPV seals properly. Therefore we can expect that id_0=0, i.e. there is no loss at zero shaft speed, regardless of outlet pressure.

The losses at full displacement and atmospheric pressure on the outlet are:

$$P_{lploss}(\omega) = \begin{bmatrix} 1 & \omega & \omega^2 & \omega^3 \end{bmatrix} \cdot \begin{bmatrix} lp_0 \\ lp_1 \\ lp_2 \\ lp_3 \end{bmatrix} \tag{22}$$

P_{lploss} - $P_{idleloss}$ represents the power required to pump the fluid through the internal flow passages of the DDP. Again we can expect lp_0=0 because at zero shaft speed, there is zero flow.

At maximum pressure, the losses are:

$$P_{hploss}(\omega) = \begin{bmatrix} 1 & \omega & \omega^2 & \omega^3 \end{bmatrix} \cdot \begin{bmatrix} hp_0 \\ hp_1 \\ hp_2 \\ hp_3 \end{bmatrix} \tag{23}$$

P_{hploss} represents the total losses from all sources. Here hp_0 can be expected to be non-zero, because of internal leakage.

These three equations define three curves:

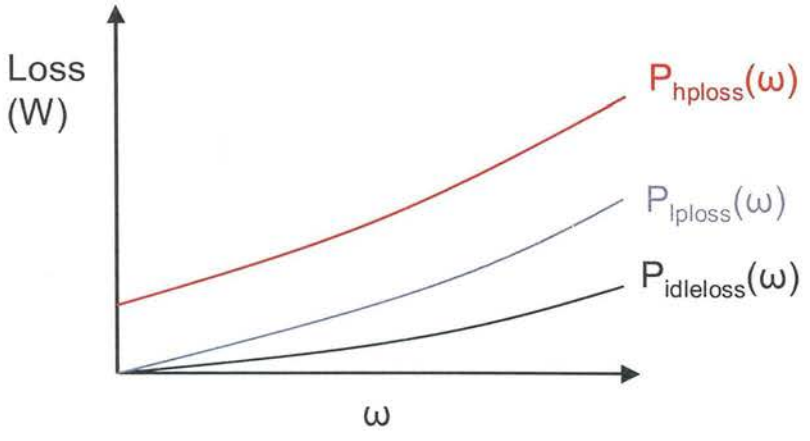


Figure 2-33: The three polynomial curves which describe losses

The losses at full displacement $P_{full\ disp.\ loss}$ (ω,p) are assumed to be proportional to pressure:

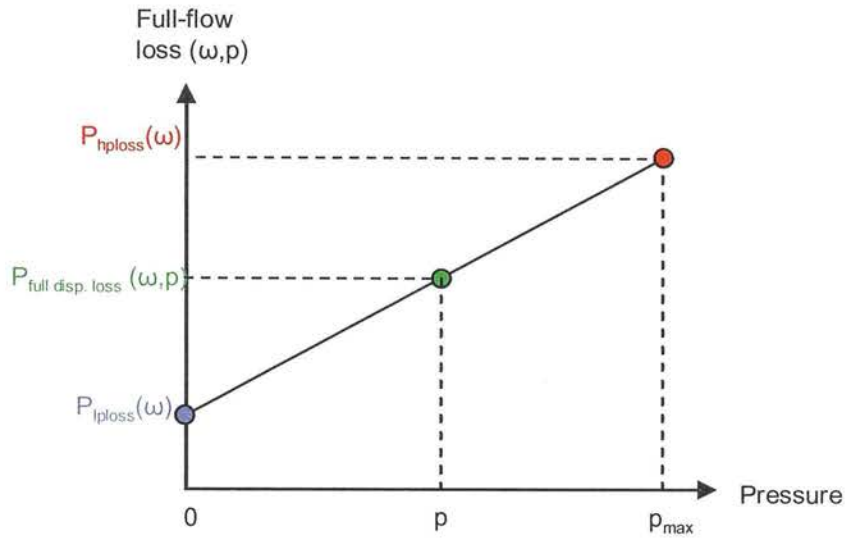


Figure 2-34: Linear interpolation of losses with respect to pressure

This then defines the curve $P_{fulldisp.loss}$ (ω,p) of losses as a function of speed at full displacement, at a given pressure p:

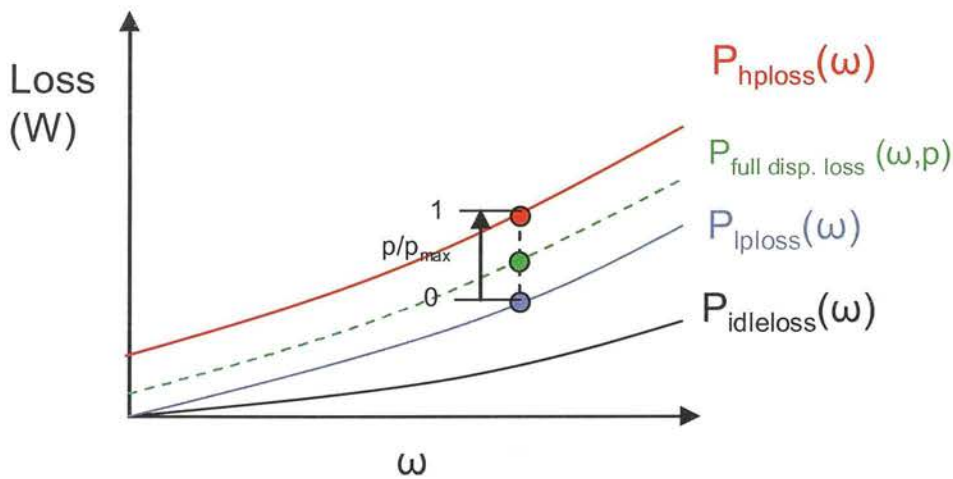


Figure 2-35: The derived curve of losses at full displacement at the pressure of interest

Consider the machine operating at a fraction F_d of full displacement. If F_d=0, all the cylinders are always in the idle state and the losses are fully defined by the curve $P_{idleloss}$. If F_d=1, all the cylinders are always in the pumping state and the losses are fully defined by the curve $P_{full\ disp\ loss}$(ω,p). If F_d is between 0 and 1, then on average F_d of the cylinders are in the pumping state and (1- F_d) of the cylinders are in the idle state:

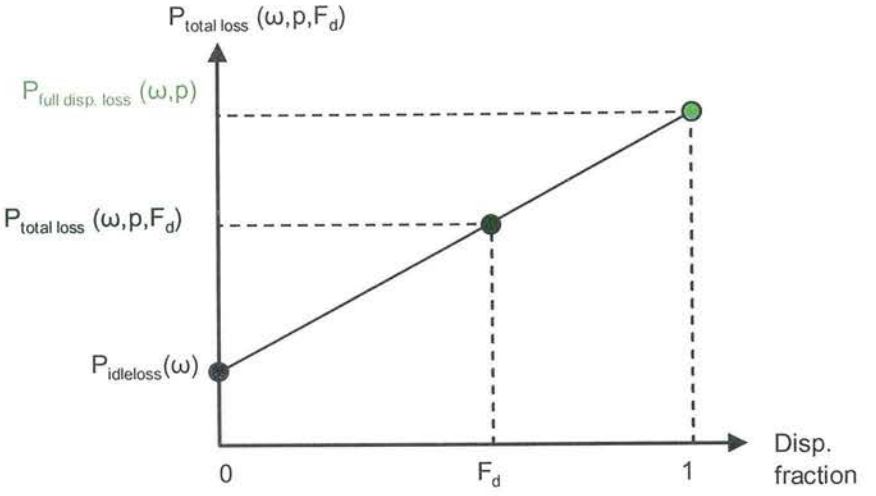


Figure 2-36: Linear interpolation of losses with respect to displacement fraction

This defines a curve $P_{total\ loss}(\omega,p,Fd)$ of losses as a function of speed at a given pressure and displacement fraction:

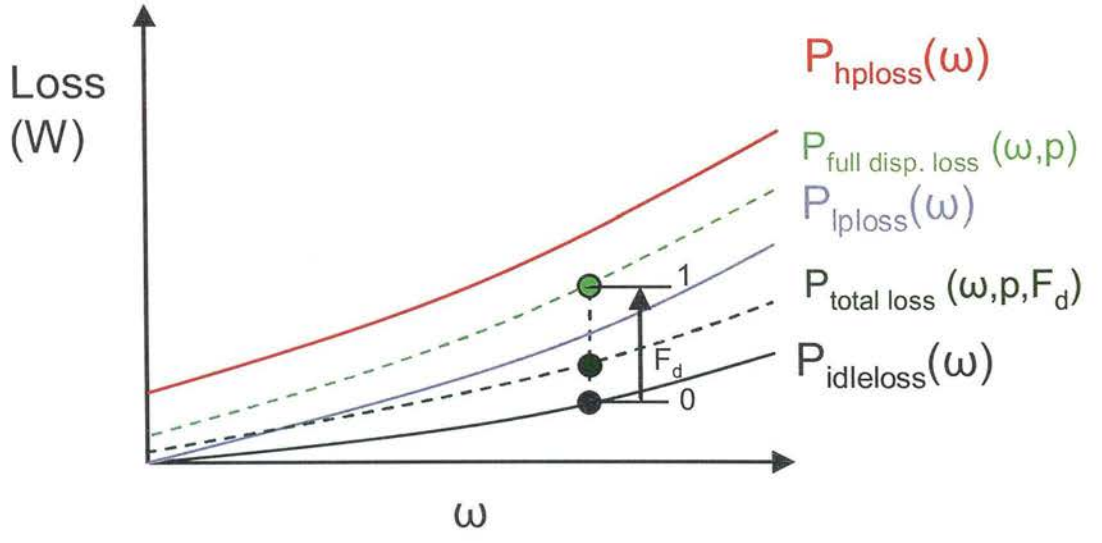


Figure 2-37: The derived curve of total losses at the pressure and displacement fraction of interest

$$P_{totalloss}(\omega) = P_{idleloss}(\omega)\cdot(1-F_d) + \left[P_{lploss}(\omega)\cdot(1-\frac{p}{p_{\max}}) + P_{hploss}(\omega)\cdot(\frac{p}{p_{\max}})\right]\cdot F_d \quad (24)$$

2.6.3 Calculating leakage

The above equations model the total losses in a DDP. A complete forward-facing model must output the torque and flow for a given shaft speed, displacement fraction and pressure. This

means that the effects of leakage and compressibility must be brought into the model. Once the output flowrate is calculated, the output power is fully defined, being the product of flow and pressure. The input power is simply the output power plus the total losses, and as the speed is specified, the torque is determined from the input power.

Considering the total loss curves defined previously, it can be seen that one of the curve coefficients holds important information about the leakage loss:

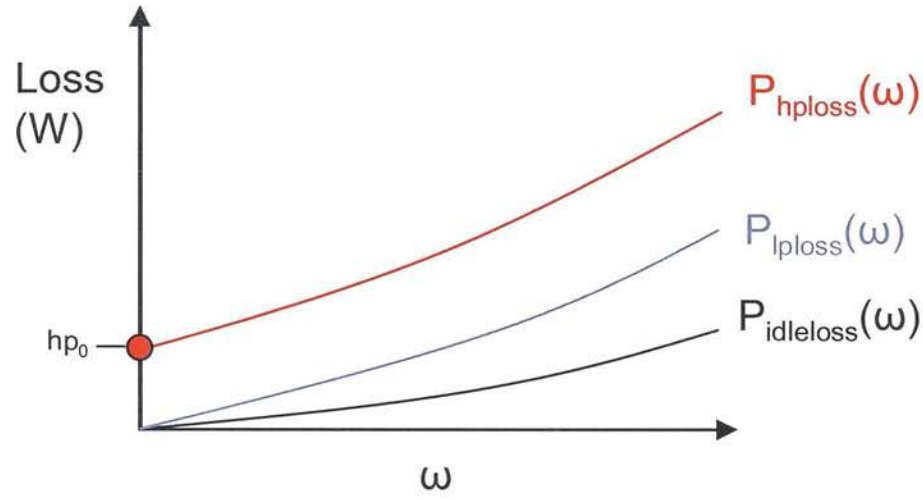


Figure 2-38: The losses at zero speed and high pressure, due solely to leakage

At zero speed, the losses at low pressure (full displacement) and at idle are zero, because the losses included in both of these curves are purely mechanical. However, the losses at high pressure are non-zero, because the pump leakage is not proportional to speed. It should be pointed out that the leakage loss at zero speed in a DDP cannot be measured directly by applying back-pressure to the port and measuring leakage flow, as this is prevented by the check-valve function of the HPVs. Instead the hp_0 coefficient is derived from extrapolating the data obtained from tests over a range of speeds (see Figure 2-42, page 62). There is some error inherent in calculating hp_0 by this method and this is dealt with by means of a correction factor, $k_{leakscale}$ below.

Assuming that hp_0 is accurately known, the leakage flow equivalent to this can be easily calculated:

$$Q_{hp0} = \frac{hp_0}{p_{\max}} \tag{25}$$

It is well known that the leakage flow in a hydrostatic machine is a function of both speed and pressure (Dorey 1988). The functions $C_{speed}(\omega)$ and $C_{pressure}(p)$ are therefore introduced to

model this, and an overall scaling factor $k_{leakscale}$ accounts for any error in the calculation of hp_0:

$$Q_{leak,fulldisp.} = Q_{hp0} \cdot k_{leakscale} \cdot C_{speed} \cdot C_{pressure} \tag{26}$$

The coefficient C_{speed} is defined as a linear function of the constant k_{speed}, being the ratio between the leakage at maximum speed to the leakage at zero speed where hp_0 is derived:

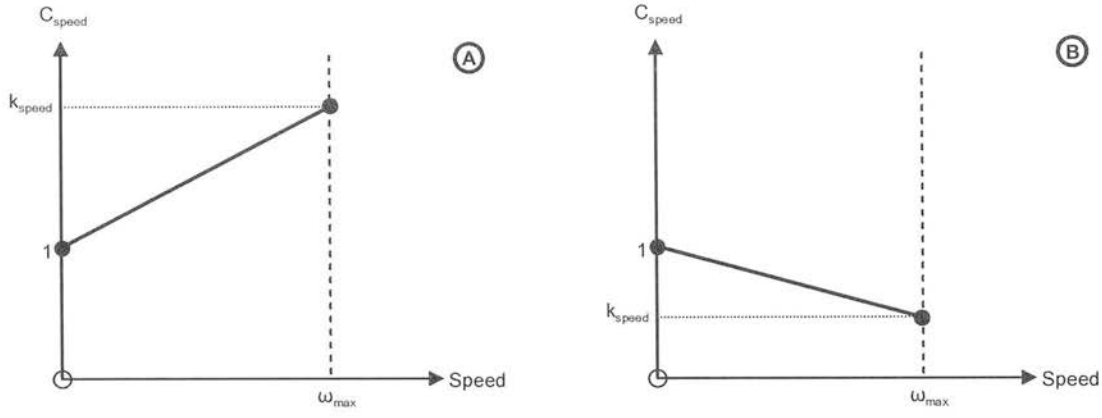


Figure 2-39: Two cases for the value of k_{speed}

$$C_{speed} = 1 + \frac{\omega}{\omega_{\max}}(k_{speed} - 1) \tag{27}$$

Case (A): $k_{speed} > 1$; the leakage rises with speed; this is the behaviour found by Dorey (1988) for a radial piston pump.

Case (B): $k_{speed} < 1$; the leakage falls with speed.

To arrive at an expression for $C_{pressure}$ let us first consider what the simplest function expected would be. If the leakage paths in a hydrostatic unit could be lumped together as a single viscous impedance of constant dimensions, then the leakage flow would be simply proportional to pressure - this is classical Poiseuille flow. In the DDP machines described in this work, the important leakage paths are the cylinder/shell spherical bearing, the piston pad bearing and the piston/cylinder clearance. The first two of these are overclamped hydrostatic/hydrodynamic bearings - the clearances of these bearings result from a dynamic force balance between hydrostatic and hydrodynamic forces. This behaviour is complex and the gap resulting cannot be assumed to be constant with pressure. The thin walls of the piston and cylinder can be expected to strain and therefore the gap between them may increase or decrease with pressure depending on their relative hoop stiffness. Indeed it has been proposed that the stiffness of the piston and cylinder gap can be designed such that the gap reduces in a

controlled manner as pressure increases, to achieve an optimum balance between shear and leakage (Salter and Rampen 1993).

To model these effects the author proposes that the Poiseuille flow assumption be modified with a normalised gap function $x_{norm}(p)$. This normalised gap is 1 at p_{max}, being the pressure for which hp_0 was calculated. The gap is modelled to either reduce or increase linearly with pressure depending on the value of p_{zero}, which is the pressure at which the gap closes completely:

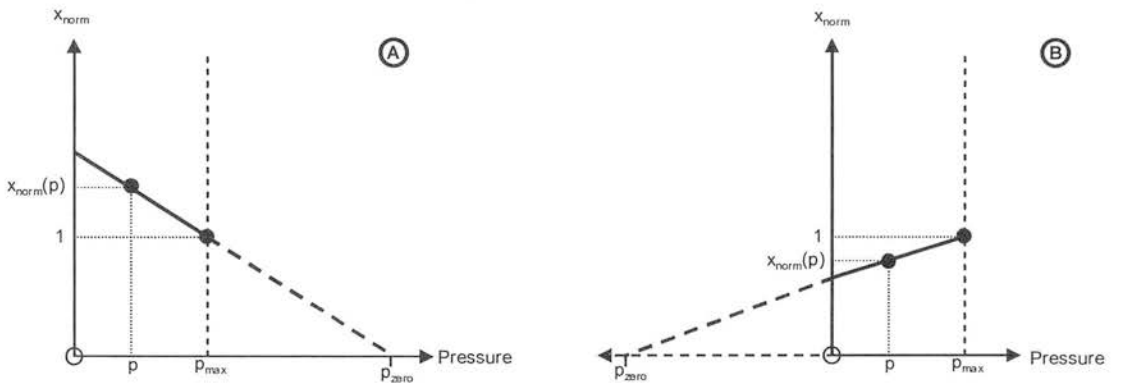


Figure 2-40: Two cases for the value of p_{zero}

$$x_{norm} = \frac{p_{zero} - p}{p_{zero} - p_{\max}} \tag{28}$$

Case (A): p_{zero} is positive and well above p_{max}; the gap closes as pressure rises such that x_{norm} is higher at $p=0$ than at p_{max}. In terms of the piston/cylinder gap, the piston is of lower stiffness than the cylinder and so the gap closes with positive pressure in the chamber.

Case (B): p_{zero} is negative; the gap opens up as pressure rises such that x_{norm} is lower at $p=0$ than at p_{max}. In terms of the piston/cylinder gap, the piston is of lower stiffness than the cylinder and so the gap closes with *negative* pressure in the chamber; i.e. when the case pressure is higher than the chamber pressure.

The leakage flow through a thin gap with viscous flow is proportional to the pressure and the third power of the gap (Douglas 1995). Therefore $C_{pressure}$ is defined:

$$C_{pressure} = x_{norm}{}^{3} \cdot \frac{p}{p_{\max}} \tag{29}$$

The effect of p_{zero} on the relationship between pressure and leakage flow is shown below.

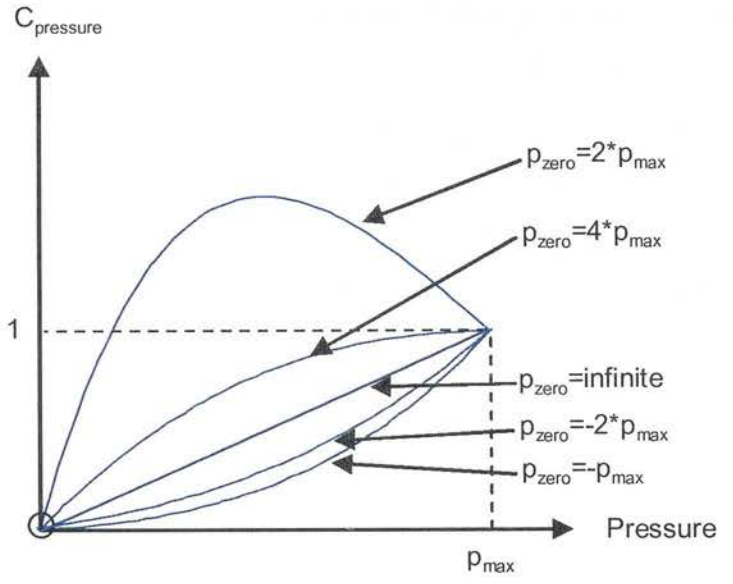


Figure 2-41: The effect of the value of p_{zero} on the modelled leakage with respect to pressure

The leakage scales linearly with displacement, because chambers which are idle do not leak:

$$Q_{leak} = Q_{leak,fulldisp.} \cdot F_d \tag{30}$$

2.6.4 Calculating output flow and input torque

The DDP has a nominal displacement V_{nom}. The nominal flow at the output of the pump, before the effects of leakage and compressibility is:

$$Q_{nom} = V_{nom} \cdot \omega \cdot F_d \tag{31}$$

The DDP has a "dead volume" inside each chamber, the size of which is determined by the factor k_{dead}, which expresses the dead volume as a fraction of the swept volume. The effective displacement of the DDP will reduce as the pressure increases; this reduction can be expressed as a flow:

$$Q_{comp} = Q_{nom} \cdot (1 + k_{dead}) \cdot \frac{p}{\beta} \tag{32}$$

The actual flow produced at the outlet of the pump is:

$$Q_{out} = Q_{nom} - Q_{comp} - Q_{leak} \tag{33}$$

With the output flow calculated, the output power is

$$P_{out} = Q_{out} \cdot p \tag{34}$$

The input power is now defined as:

$$P_{in} = P_{out} + P_{totalloss} \tag{35}$$

giving the input torque:

$$T_{in} = \frac{P_{in}}{\omega} \tag{36}$$

2.6.5 Calculating volumetric and mechanical losses

The commutation of the working chambers of the DDP is achieved by self-acting valves, therefore some of the strain energy stored in the compressed oil may be returned to the shaft after TDC before the LPV passively re-opens. Therefore the flow missing from the outlet due to compressibility is not assumed to comprise a volumetric power loss, because that flow is not throttled to case drain but expands to do work against the crankshaft after TDC. Therefore the volumetric power loss is due to the leakage losses alone:

$$P_{lossvol} = Q_{leak} \cdot p \tag{37}$$

The mechanical power loss follows:

$$P_{lossmech} = P_{totalloss} - P_{lossvol} \tag{38}$$

This mechanical power loss is usually presented as a torque:

$$T_{loss} = \frac{P_{lossmech}}{\omega} \tag{39}$$

The DDP loss model is now complete.

2.7 Modelling SD1B results

2.7.1 Losses at full displacement

Calculation of the model parameters starts by considering the curves of total loss vs. speed. Three 3rd-order polynomial curves describe the relationship between observed losses and speed in three conditions:

$P_{idleloss}(\omega)$ – the curve of loss in the idle condition.

P_{lploss} (ω) – the curve of loss at full displacement and minimal pressure.

P_{hploss} (ω) – the curve of loss at full displacement and maximum pressure (p_{max}).

For modelling the SD1B, p_{max} was chosen to be 300 bar – the maximum pressure used during the tests. The coefficients of P_{hploss} , P_{lploss} and $P_{idleloss}$ were derived by fitting 3rd order polynomial curves to loss vs. speed results, for tests 121 (p=300 bar) , 101 (p<5bar) and 100 (idle) respectively.

The figure below shows the output of the model compared with the experimental results:

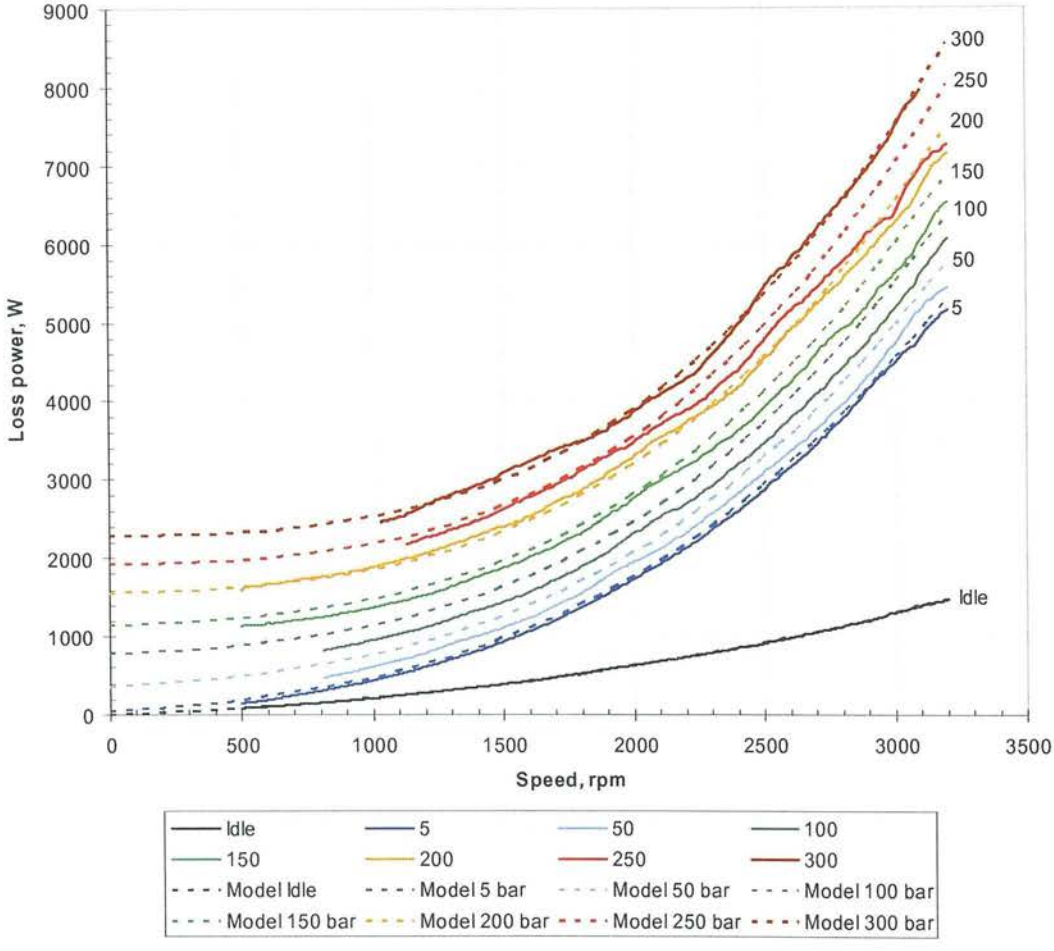


Figure 2-42: SD1B comparison between model (dashed) and experiment (solid); losses vs. speed; lines of pressure (bar)

In the figure above, the "Model Idle" "Model 5 bar" and "Model 300 bar" dashed curves are the polynomials input to the model described above. The "Model 50", "Model 100", "Model 150", "Model 200" and "Model 250" are the results from the model which assumes that the losses scale linearly with pressure. Overall, 3rd order polynomials adequately match the measured data.

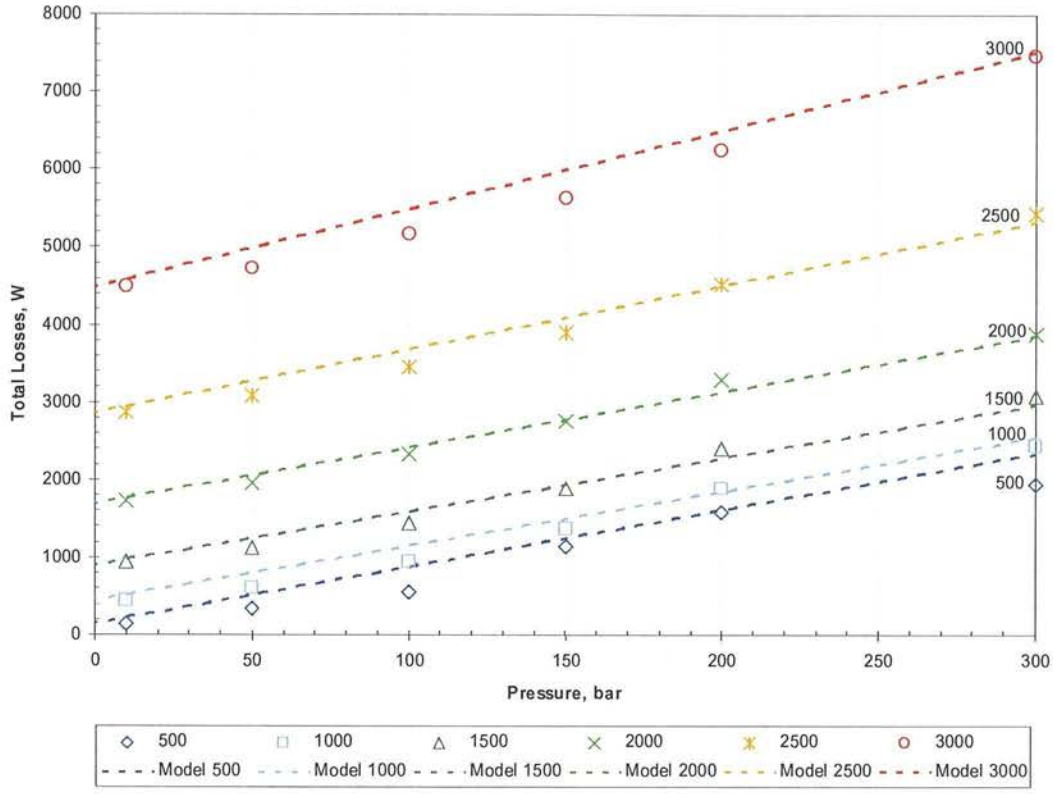


Figure 2-43: SD1B comparison between model (dashed) and experiment (points); losses vs. pressure; lines of speed (rpm)

The assumption of a linear relationship between pressure and losses is surprisingly good given the underlying complexity of the loss sources.

2.7.2 Flow and torque losses

The leakage is modelled as a gap with Poisselle viscous flow, in which the gap is assumed to linearly change with pressure. Here the important parameter is p_{zero} which is the pressure at which the gap is closed up completely. This determines the degree of curvature of the leakage function above. With p_{zero}=infinity, there is no closing of the gap and the leakage is linear with respect to pressure. With a negative value of p_{zero}, the gap opens up with pressure.

For the SD1B data, a reasonable match between modelled and observed leakage behaviour with respect to pressure, was achieved with p_{zero} = 1200 bar, as shown below.

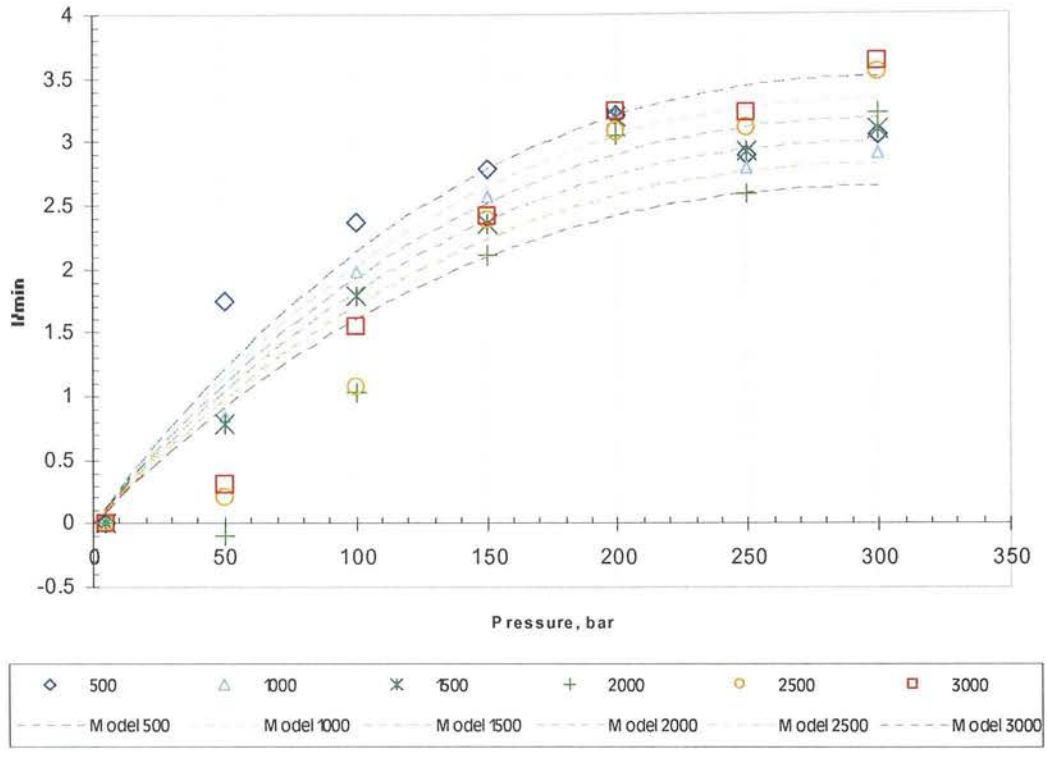


Figure 2-44: SD1B comparison between model (dashed) and experiment (points); leakage flow (l/min) vs. pressure (bar); lines of speed (rpm)

The model parameter k_{speed} describes a linear relationship between leakage and speed. For the SD1B data, a reasonable match between modelled and observed leakage behaviour with respect to speed, was achieved with $k_{speed} = 0.7$, as shown below.

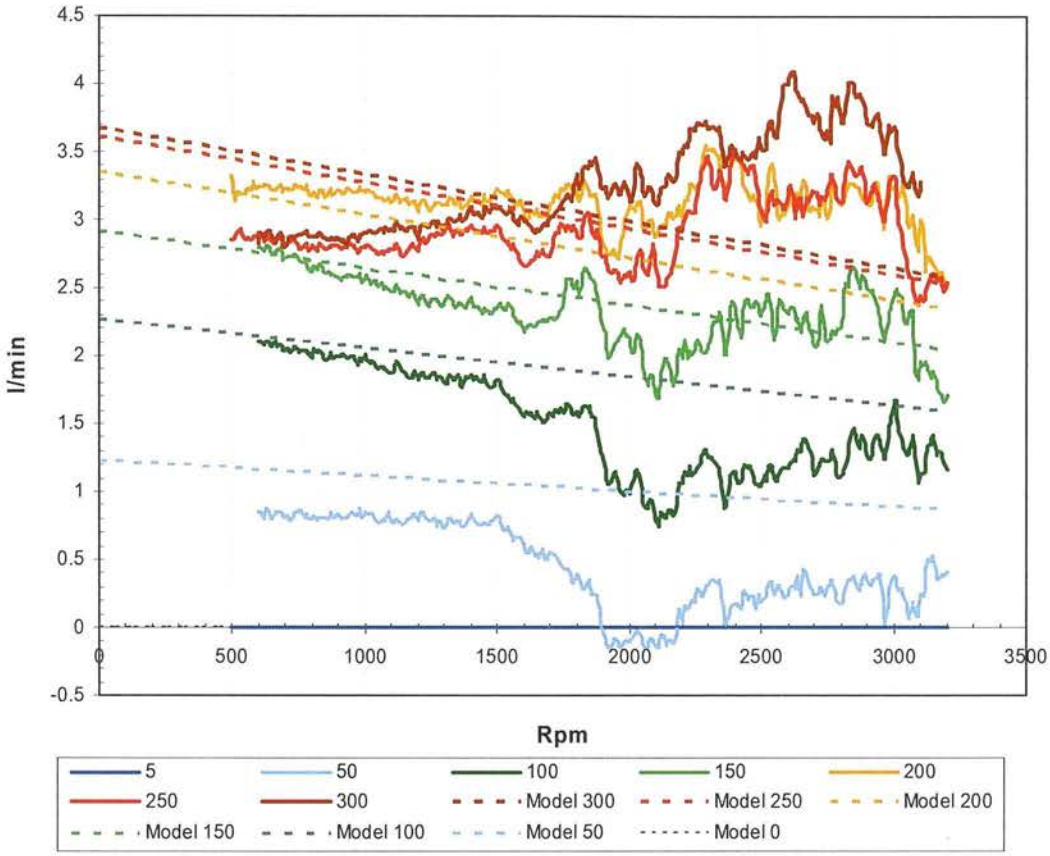


Figure 2-45: SD1B comparison between model (dashed) and experiment (solid); leakage flow (l/min) vs. speed (rpm); lines of pressure (bar)

The results above show the modelled leakage flow with the parameter $k_{speed} = 0.7$, meaning that at maximum speed the leakage flow is modelled to decrease to 0.7 of the amount at zero speed. This was the best match that could be obtained, but obviously there is a lot more going on in the real machine than is captured by the model; there is potential here for further study.

Once the total losses and the leakage losses are modelled, the mechanical losses are derived by subtraction and converted into a torque loss. The results of this are shown below.

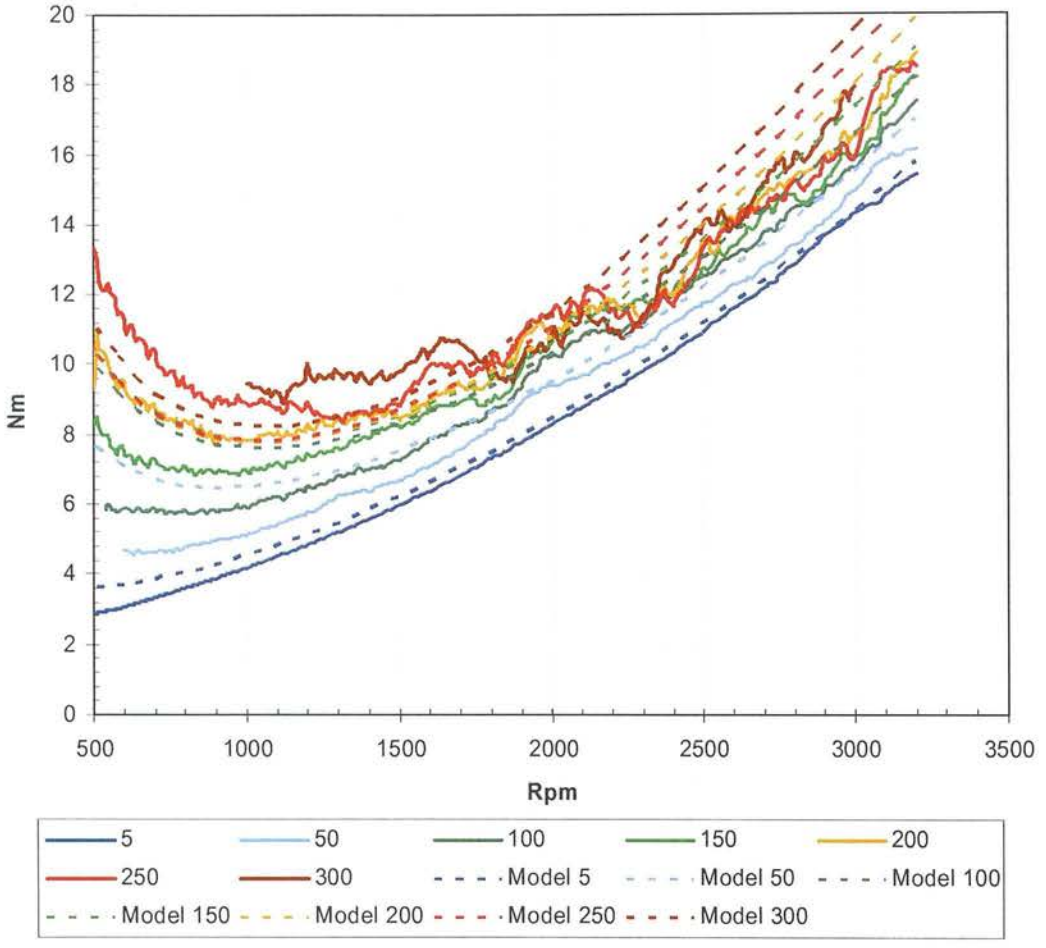


Figure 2-46: SD1B comparison between model (dashed) and experiment (solid); torque loss vs. speed (rpm); lines of pressure (bar)

Despite the simplistic model of leakage, the real behaviour is reproduced reasonably well. At zero pressure, all of the observed total losses are attributed to torque loss. At low speeds the torque loss increases, probably due to increased friction from a reduced hydrodynamic lubrication effect – in this zone the effect of pressure is magnified. At high speeds the torque loss is dominated by the flow-related pressure drop inside the flow passages and is only weakly affected by pressure.

2.7.3 Effective displacement

With the leakage and mechanical losses calculated, the output flow is calculated by the model assuming that the effect of compressibility is to linearly reduce the displacement according to eq. 33 (p.61). Setting the parameters k_{dead} =2.80 and β=18000 bar yields the results shown below:

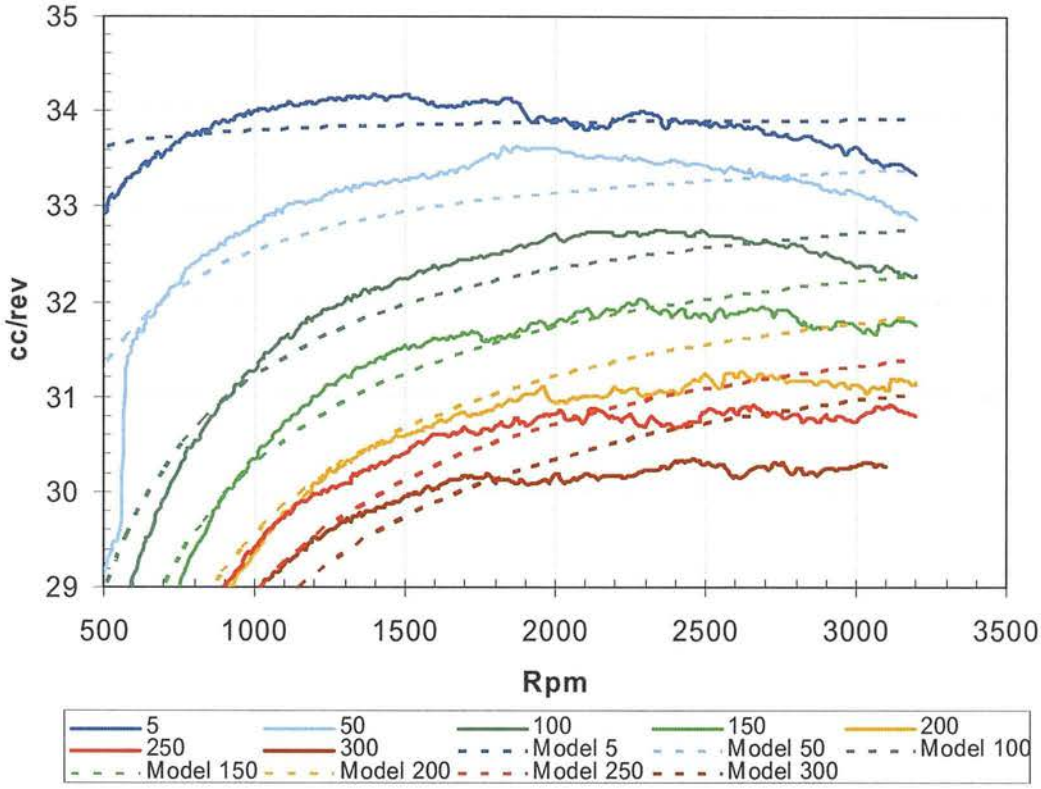


Figure 2-47: SD1B comparison between model (dashed) and experiment (solid); V_{eff} (cc) vs. speed (rpm); lines of pressure (bar)

The effects of compressibility and leakage are captured fairly well; however the variation in V_i, due to the valve timing and other effects, is not modelled.

2.7.4 Efficiency

With the total losses and the output flow modelled, the overall efficiency may be calculated from the model, as shown below:

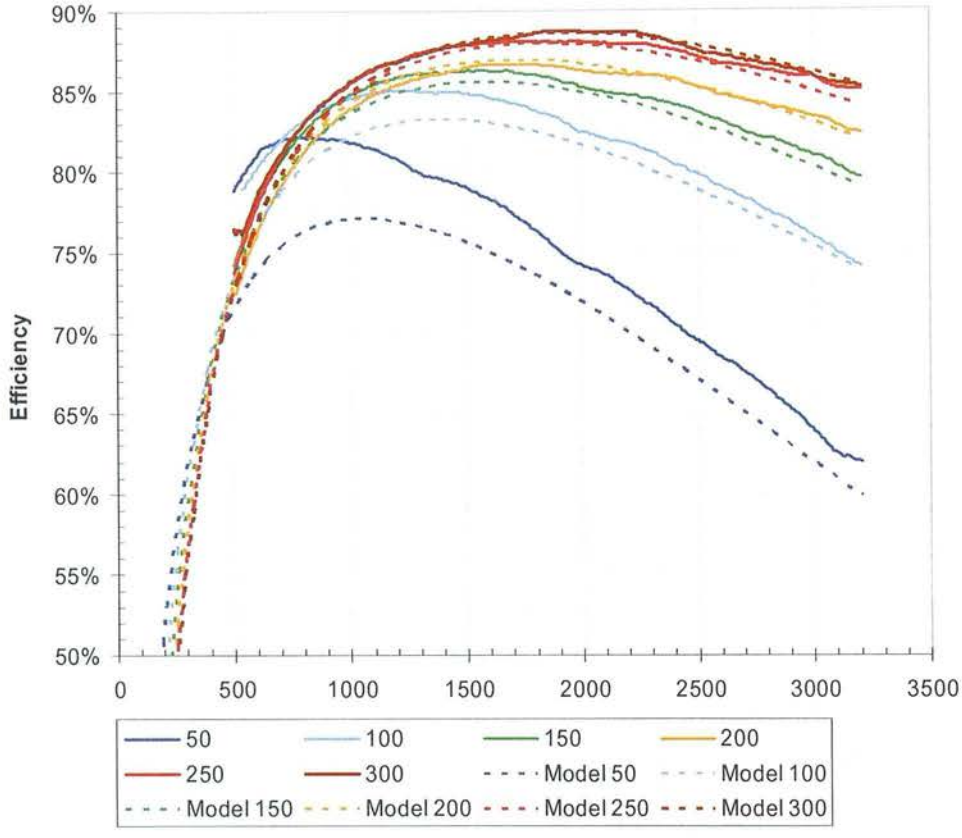


Figure 2-48: SD1B comparison between model (dashed) and experiment (solid); overall efficiency vs. speed (rpm); lines of pressure (bar)

The model matches the actual overall efficiency results fairly well and certainly exhibits the same form. At 150 bar and above, the error is around 1%, whilst the maximum error is around 6% at p=50 bar and speed=500rpm.

2.7.5 Summary of coefficients

The model parameters which were derived to match the SD1B data are given below:

Table of loss coefficients:

Order	Idle	LP	HP
0	0.000E+00	0.000E+00	2.300E+03
1	1.427E+00	1.936E+00	2.023E-01
2	5.022E-03	1.273E-02	2.755E-03
3	1.147E-05	8.351E-05	1.560E-04

Constants:

V_{nom}=34cc p_{max}=300 bar k_{dead}=2.80 β=18000 bar

p_{zero}=1200 bar $k_{leakscale}$=0.8 k_{speed}=0.7 n_{max}=3200rpm

2.8 The C2 test machine

The SD1B results were useful to validate the DDP efficiency model, but the magnitude of overall efficiency demonstrated was disappointing. Compared with axial piston machines, the peak efficiency of 88.5% is respectable but not remarkable. It was felt that the efficiency could be substantially improved with some design changes. This resulted in the creation of the "C2" machine by Gordon Voller and Uwe Stein of AIP.

The C2 was of similar overall design to the SD1B, but had only one bank of 6 pistons giving a total displacement 12cc/rev. Compared to the SD1B, it had the following improvements:

- The radial flow passages in the LPV shell were larger; investigation showed that inadequate flow area here was the main reason for the high internal pressure drop in the SD1B.
- The HPV pressure drop was reduced by increasing the flow area.
- The flow area of all the internal flow galleries was increased.
- The overclamp of the spherical cylinder hydrostatic bearing was increased to 5.5% to reduce leakage.
- The crankshaft was given a salt-bath nitriding and polishing treatment to reduce the friction coefficient.
- Aluminium slugs filled the majority of the dead volume inside the hollow pistons, reducing k_{dead} to 2.20.
- The end plate was re-designed to prevent the extrusions of the seals shown in Figure 2-31, to eliminate the possibility of parasitic leakage.

The results of tests on the C2 are shown below.

2.8.1 C2 efficiency results

The C2 was tested over the following range:

Speed = 600, 900, 1200, 1500, 1800, 2100, 2400, 2700 rpm

Pressure = <5, 50, 100, 200, 300, 400 bar

Displacement = idle, 20%, 40%, 60%, 80%, 100%

The overall efficiency results at full displacement are shown below:

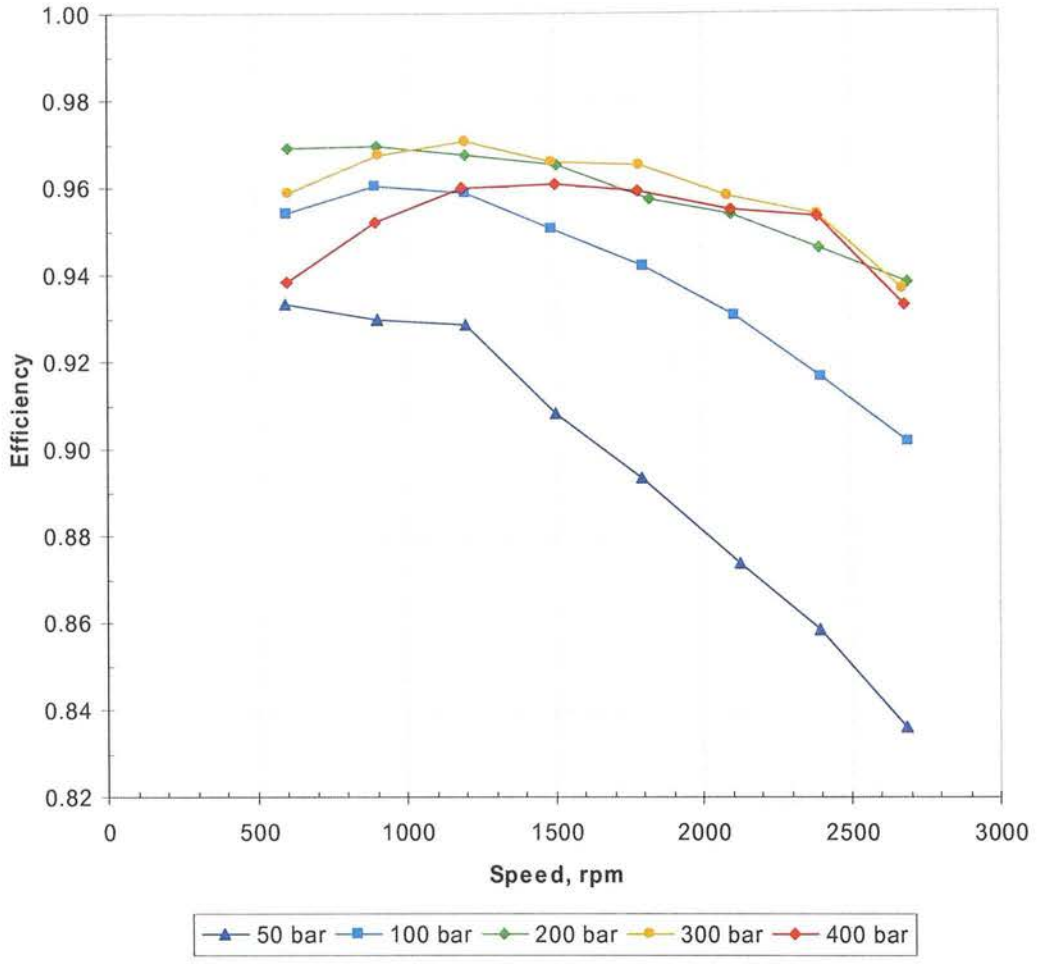


Figure 2-49: C2; Overall efficiency vs. speed (rpm) at full displacement; lines of pressure (bar)

The C2 displayed greatly improved efficiency compared to the SD1B. The peak efficiency of 97% is exceptionally high for a hydraulic pump.

The volumetric and mechanical efficiencies shown below were calculated using the standard definitions in ANSI/NFPA T.3.9.17.

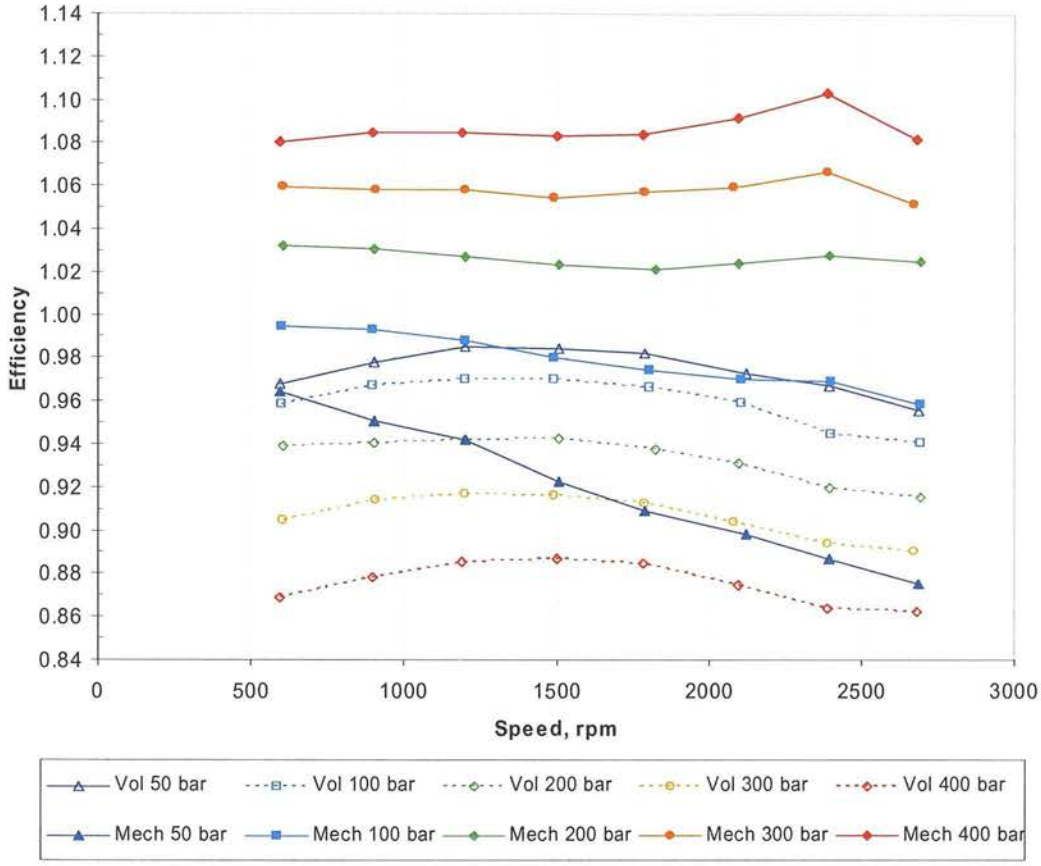


Figure 2-50: C2; volumetric and mechanical efficiencies at full displacement, by standard definitions

This figure shows very clearly that the standard definitions for volumetric and mechanical efficiency are not useful when analysing DDPs, as discussed in Section 2.4. A mechanical 'efficiency' of 110% is clearly meaningless.

The figure below shows the results of using the revised definitions presented in Section 2.4.

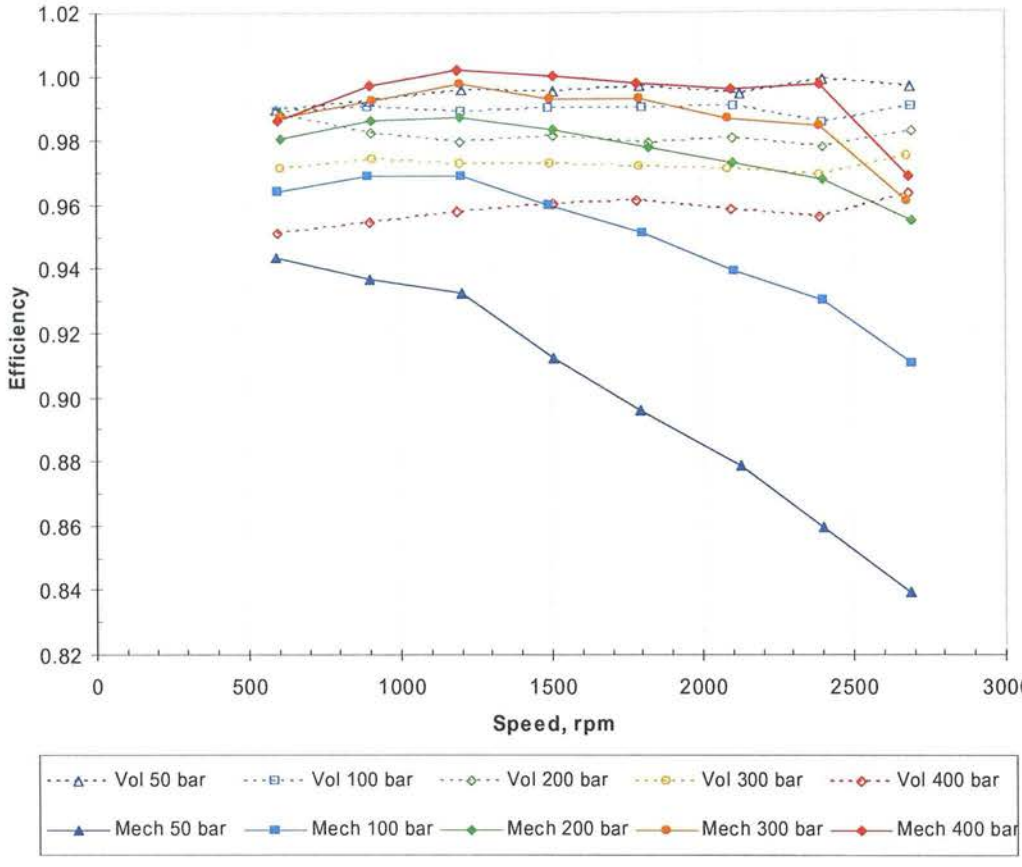


Figure 2-51: C2; volumetric (dashed) and mechanical (solid) efficiency at full displacement, by revised definitions

Clearly, the revised definition for volumetric efficiency gives more meaningful results than the standard definition.

At 1000rpm and 300 bar, the C2 had a volumetric efficiency of 97%, compared with 90% achieved by the SD1B. This indicates that the attempts to reduce leakage sources were successful.

At 50 bar, 2500rpm the mechanical efficiency of the C2 was 85%, compared with 70% achieved by the SD1B. This indicates that the attempts to reduce the internal pressure drop was also successful.

2.9 Modelling C2 results

2.9.1 Summary of model parameters for the C2

The model parameters below for the C2 were derived in a similar way to that described for the SD1B in section 2.7.1. The notable differences were that k_{dead} for the C2 was reduced to 2.20 (by filling the dead volume in the hollow pistons with aluminium plugs); the leakage was observed to increase strongly with speed (k_{speed}=3); and the leakage was more linear with respect to pressure (p_{zero}=2000 bar).

Table of loss coefficients:

Order	Idle	LP	HP
0	0.000E+00	0.000E+00	1.000E+02
1	1.500E-01	2.100E-01	2.100E-01
2	3.380E-03	3.380E-03	3.380E-03
3	1.300E-06	3.200E-06	1.800E-05

Constants:

V_{nom}=11.8cc p_{max}=300 bar k_{dead}=2.20 β=18000 bar

p_{zero}=2000 bar $k_{leakscale}$=1 k_{speed}=3 n_{max}=3000rpm

2.9.2 Comparison of C2 experimental and model results

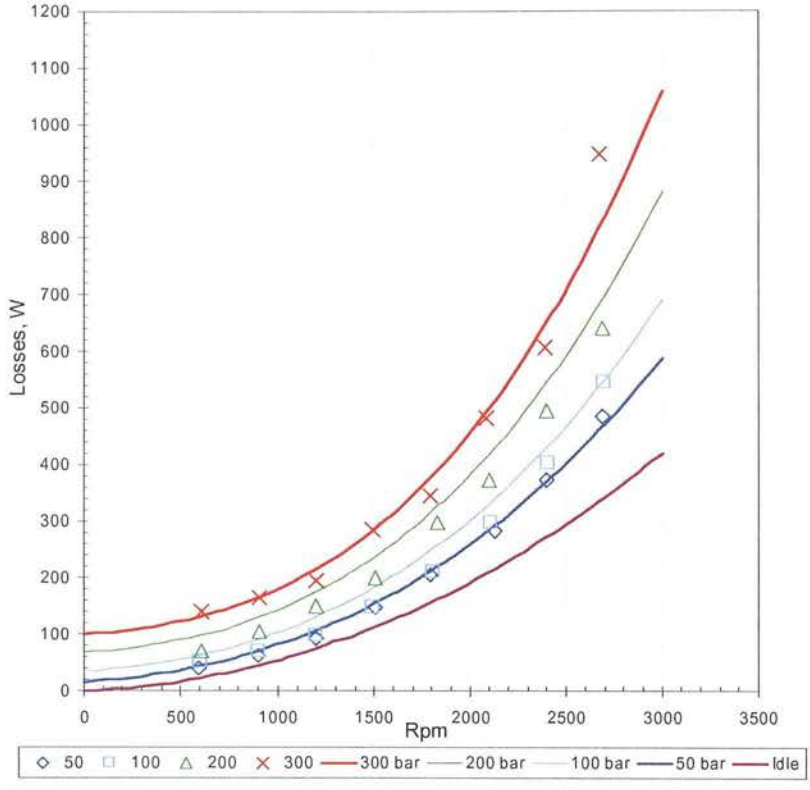


Figure 2-52 C2 comparison between model (solid) and experiment (points); losses (W) vs. speed (rpm); lines of pressure (bar) and idle (purple)

The third-order loss model seems to capture the observed behaviour fairly well.

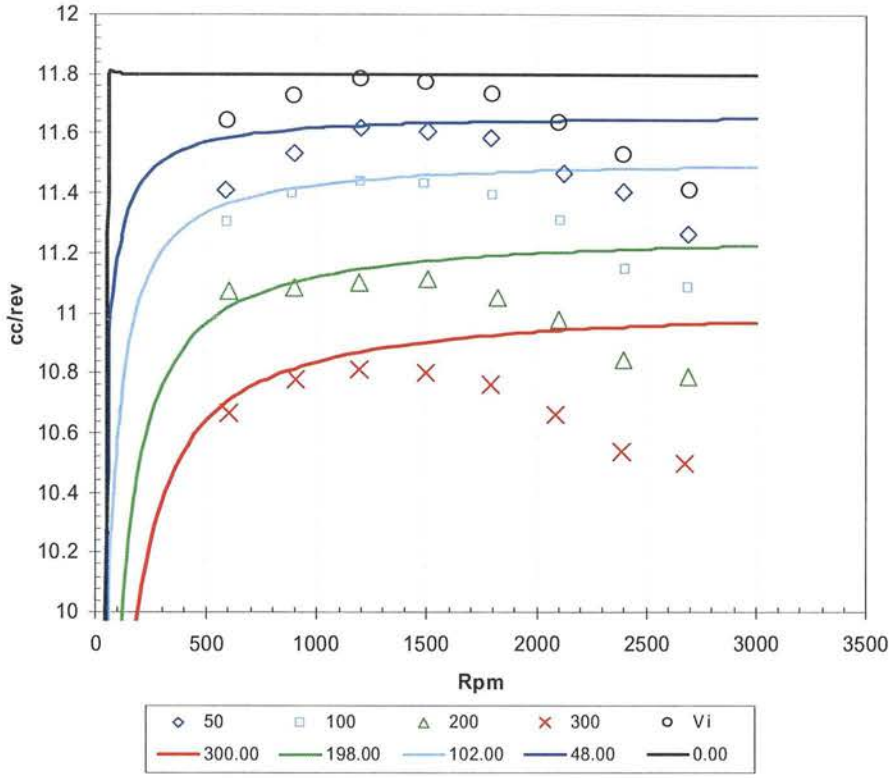


Figure 2-53: C2 comparison between model (solid) and experiment (points); effective displacement (cc) as a function of speed (rpm); lines of pressure (bar)

There is more pronounced speed-related variation of displacement with the C2 than the SD1B; this could be due to variations in the commutating valve timing. There is no attempt in the model to match this. An improvement to the model could be made by defining V_{nom} to be a function of speed, perhaps implemented as a look-up table.

The output of the model for overall efficiency is compared to experimental results below:

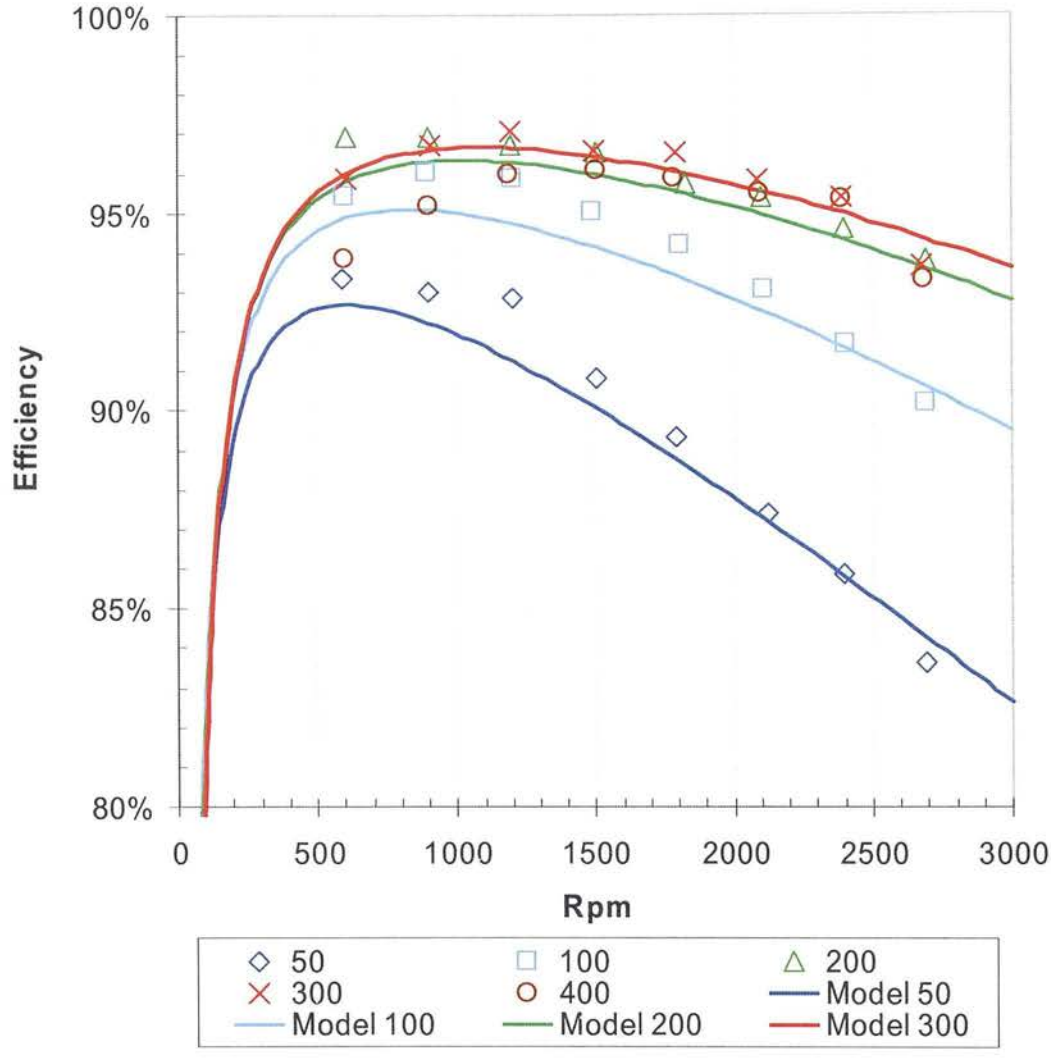


Figure 2-54 C2 comparison between model (solid) and experiment (points); overall efficiency as a function of speed (rpm); lines of pressure (bar)

Overall, the model accurately matches the form of the experimental results. At 200 bar and 300 bar, the model is accurate to within 1%; the maximum error of 1.8% occurs at 50 bar, 1200rpm.

2.9.3 Comparison with a swashplate pump at full displacement

Dorey (1988) presents a model of torque and flow losses of an axial-piston machine as a function of speed, pressure and displacement fraction. Usefully, he also discloses the actual parameter set for a particular machine under test – a 161cc axial piston swashplate pump. These equations and parameters were used to re-create Dorey's model; the overall efficiency is compared below to the C2 results.

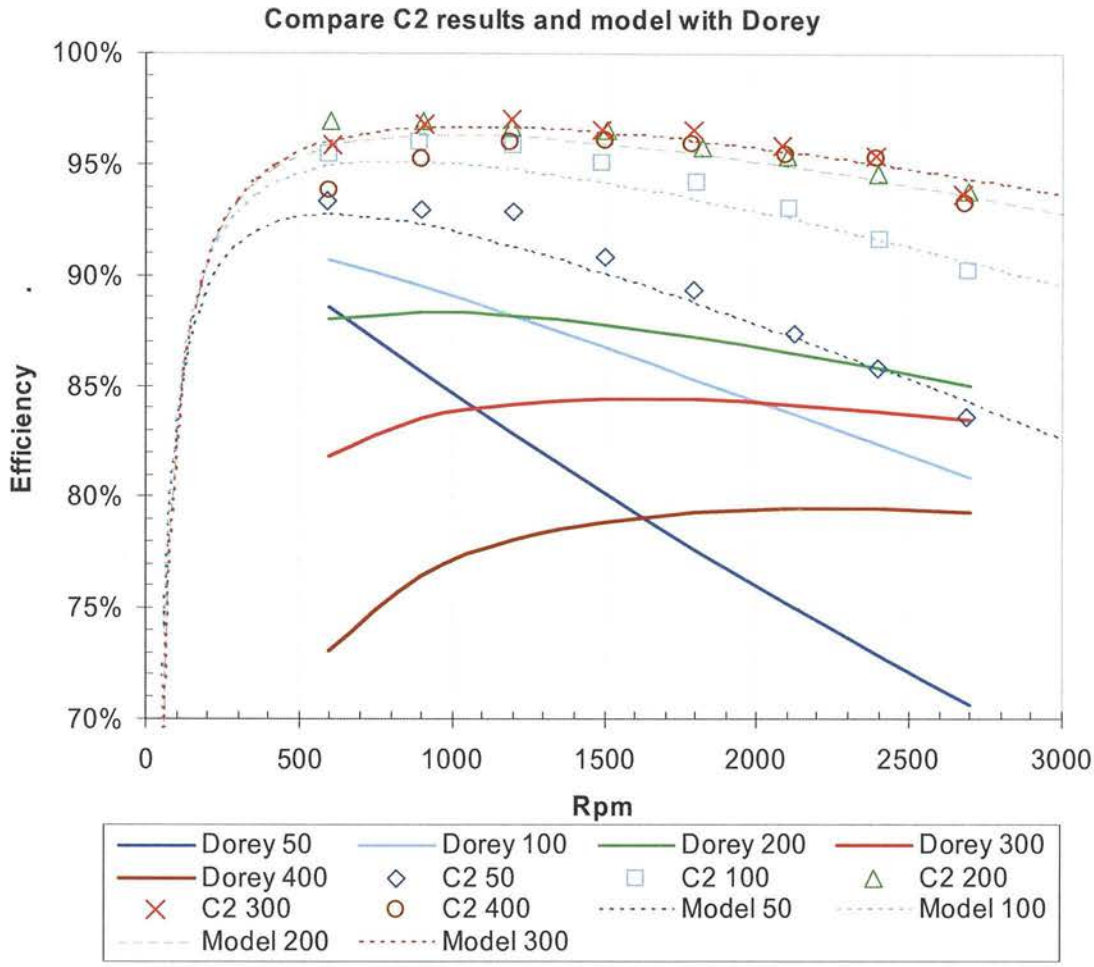


Figure 2-55: Comparing the C2 experiments (points) and model (dashed) with Dorey's model of an axial piston pump (solid); overall efficiency at full displacement vs. speed.

At full displacement, the C2 is much more efficient over the entire operating map than the swashplate pump modelled by Dorey. Comparing the C2 experimental data with the swashplate pump model, the greatest percentage differences are seen at high speed with low pressure, and low speed with high pressure. At both 2700rpm and 50 bar, and at 600rpm and 400 bar, the C2 overall efficiency is over 20% higher than the swashplate machine, but these are both low-power conditions.

At more common operating points, the advantage is smaller but still very significant. At 200 bar, 2000rpm, the efficiency of the C2 is 95.5%, while the swashplate machine is 87%, a difference of 8.5%.

If it is assumed that a DDP, which was scaled to the size of the 161cc swashplate machine, would exhibit the same efficiency as the 12cc machine tested, then the efficiency comparison above can be considered in terms of power losses, at the 161cc scale of the Dorey machine. At 2000rpm and 200 bar, the losses of the swashplate machine would be 14.9% of the output

power, whilst the losses of the scaled C2 would be 4.7% of the output power. At this operating point, the scaled C2 would have 31.5% (i.e. less than one third) of the losses of the swashplate machine.

2.9.4 Comparison with the swashplate pump at partial displacement

It is difficult to draw conclusions about the relative levels of loss between axial-piston and DDPs at small displacement fractions because of the lack of published data about how axial-piston pumps behave at very low swash angles.

Typically the volumetric losses in variable swashplate machines remain constant as displacement reduces, while the torque losses reduce substantially as displacement fraction reduces – typically to 25%-50% of those at full displacement (McCandlish and Dorey 1984)

Dorey tested his machine at four discrete fractions of full displacement, namely 0.31, 0.53, 0.8 and 1.0. From these points he derived the function relating torque loss to displacement fraction by drawing a straight line through his data points. He arrived at the conclusion that only 4.8% of the total torque loss is fixed with respect to displacement fraction; the remaining 95.2% is proportional to displacement fraction. This seems to contradict his statement that the torque losses at zero displacement may be expected to be between 25% and 50% of those at full displacement. There are many sources of torque loss, e.g.:

Pressure- and displacement-insensitive:

- Churning inside the case
- Rolling bearing viscous loss
- Coulomb friction from shaft seals etc.

Pressure-sensitive:

- Port plate shear
- Piston pad shear
- Rolling bearing friction loss

Displacement-sensitive:

- Piston/cylinder viscous shear
- Internal pressure drops in the flow passages

Of these, only the last two can be expected to be a function of displacement fraction.

An inspection of Dorey's published test data against his model shows that his line fit is not conclusive at low displacement fractions:

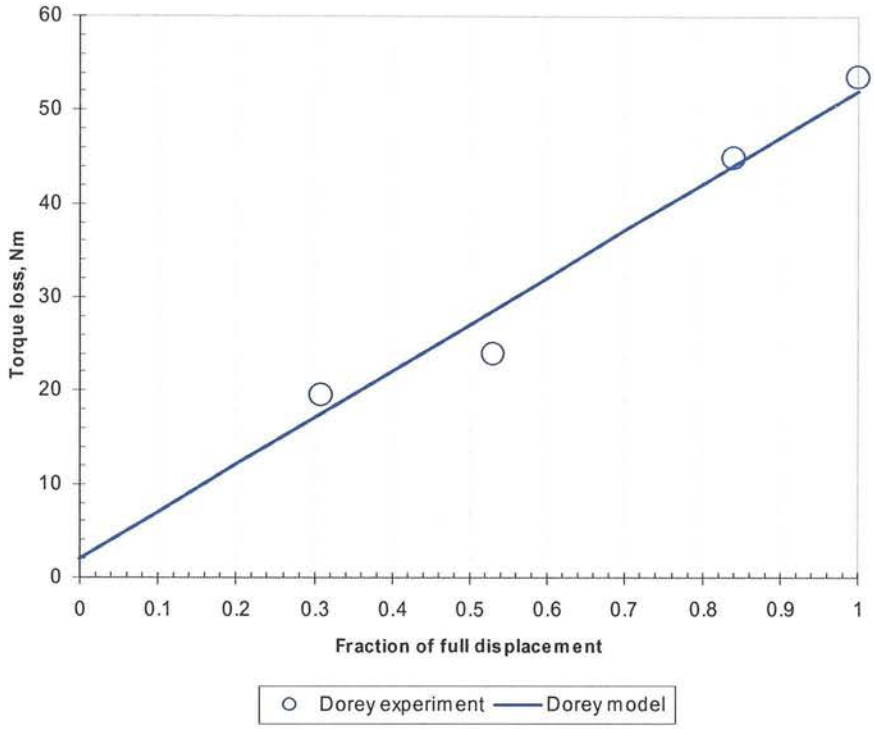


Figure 2-56: Dorey's experimental data (points) and model (line) of the torque losses of a 161cc swashplate pump at 200 bar, 1500rpm

Dorey fitted a line to 4 points in each data set to arrive at a linear relationship between torque losses and displacement. However the underlying function may not be linear and the deviations from the line suggest this. An alternative interpretation would be to create a fifth point at zero displacement (by extrapolating the line through the 0.31 and 0.53 points), and calculate the torque losses by linearly interpolating between these points.

A look-up table was added to Dorey's model to replace his linear model of the relationship between displacement and torque loss. The figure below shows the output of the model with this modification compared to Dorey's experimental data:

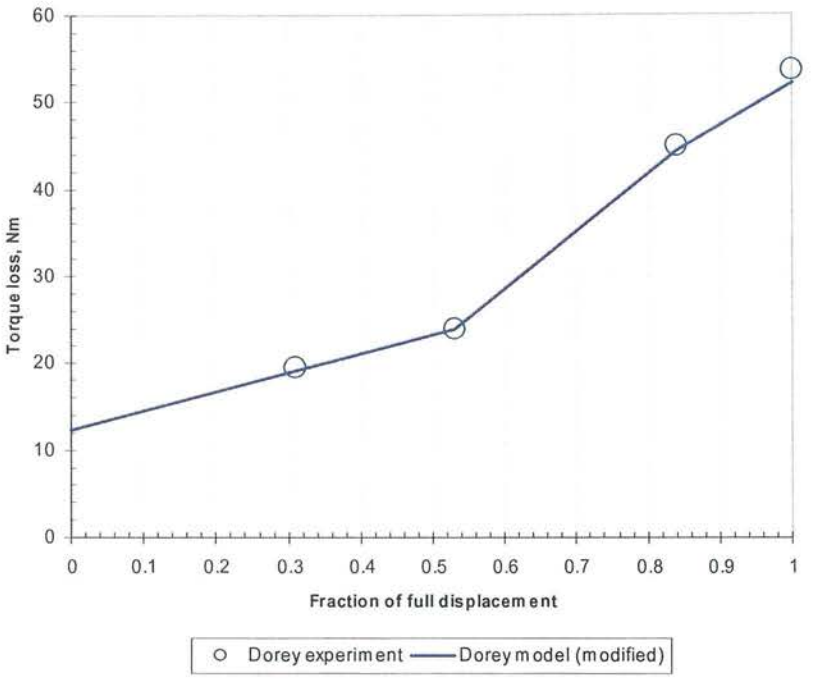


Figure 2-57: Alternative derivation of the relationship between displacement and torque losses in a swashplate pump; comparing Dorey's data to Dorey's model modified with a lookup table relating displacement to torque losses.

Using the modification above has a significant effect on the efficiency calculated from the Dorey model, at small displacement fractions.

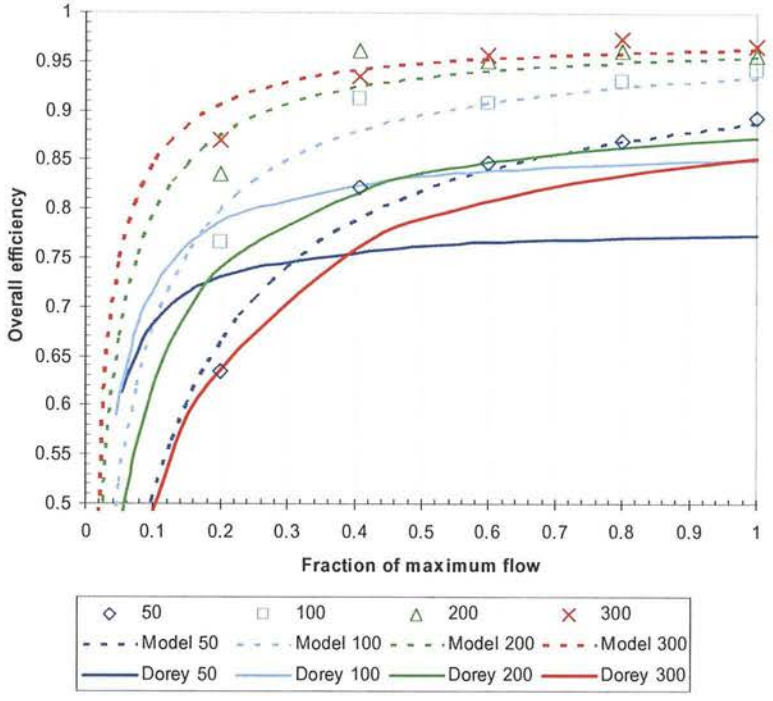


Figure 2-58: Comparison of the C2 experiments (points) and model (dashed) with Dorey's model, overall efficiency vs. displacement fraction, 1800rpm

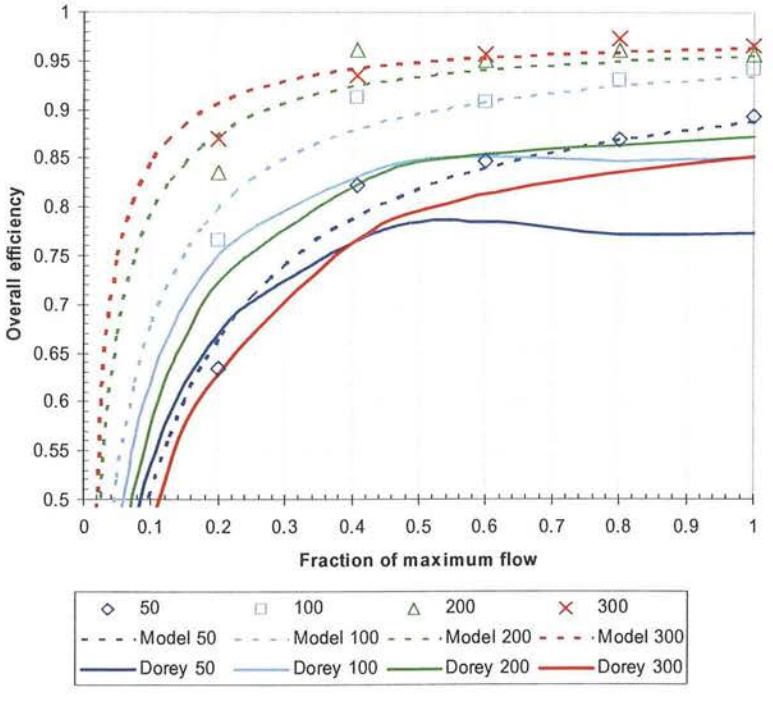


Figure 2-59: Comparison of the C2 experiments (points) and model (dashed) with Dorey's model (modified), overall efficiency vs. displacement fraction, 1800rpm

With Dorey's original model, the axial piston machine is superior at 50bar and 100 bar at very low displacement fractions. With the author's modification, the DDP has higher efficiency at all displacement fractions and pressures. It is interesting that this modification also results in the peak efficiency at low pressure for the swashplate machine occurring at less than full displacement; this phenomenon has been reported in the literature (Petersen et al. 1971).

Another way to compare the machines is to calculate the total losses as a function of output flow, assuming that a "scaled-up" 161cc DDP has the same efficiency characteristics as the 12cc C2. It is interesting to compare this with data published by manufacturers.

Some manufacturers of open-circuit pumps publish data for the input power required to maintain pressure with zero flow delivery; this is sometimes called the "high-pressure standby" condition. Is should be noted that in this condition the pump must supply enough flow to make up its own leakage losses; therefore the swashplate will be at some non-zero small fraction of its maximum pumping angle.

Parker-Hannifin Corp. (Parker 2006) manufacture the P3 series of open-circuit swashplate pump at 145cc displacement. From their literature, a 145cc unit at 1800rpm, 300 bar and 100% displacement, has an output power of 140kW at 90% overall efficiency, giving 14kW of loss. At the zero delivery condition, the input power required is 19kW. Linearly scaled to 161cc, these would be 15.1kW and 20.5kW respectively.

A similar analysis was carried out on the Vickers PVB90 191cc pump made by Eaton Corp. (Eaton 2006). These results are compared below with the results of the modified Dorey model and the C2 results. In all cases the losses are linearly scaled to 161cc displacement to allow direct comparison.

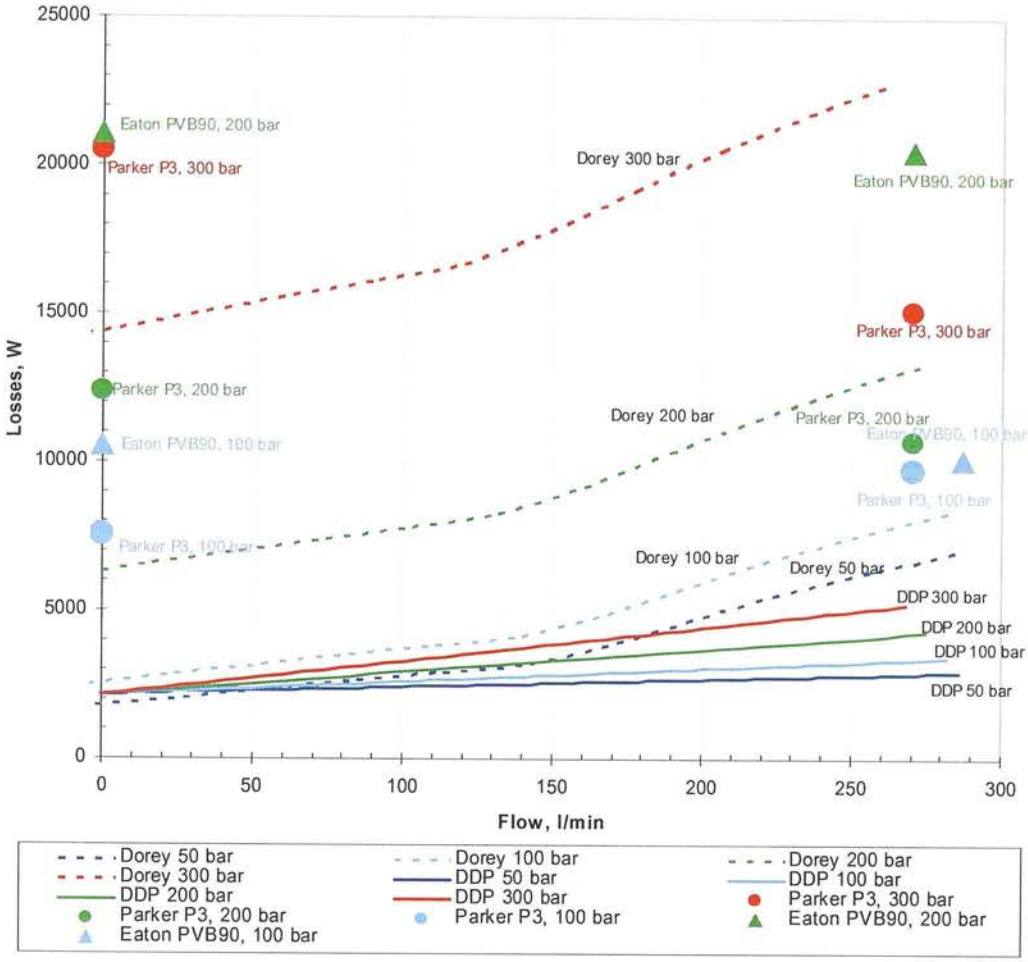


Figure 2-60: Comparing the C2 model (solid) with the modified Dorey model of an axial piston pump (dashed) and manufacturers' data (points); losses (W) at 161cc scale vs. output flow (l/min); 1800rpm.

It is notable that even with the modification described above, the losses predicted by the Dorey model at zero output flow are significantly lower than those published for the Parker and Eaton machines.

Comparing the scaled C2 to the modified Dorey model, at 300 bar the DDP exhibits less than one quarter of the losses across the displacement range. At 200 bar the DDP exhibits around one third of the losses across the displacement range. At low pressure and low flow, the losses are similar.

Comparing the scaled C2 results to the Parker P3, at 300 bar and full displacement the DDP losses are around one third, while in the high-pressure standby condition the DDP losses are less than one fifth.

Against the Eaton PVB90, the DDP has less than one fifth of the losses at full flow at 200 bar, while at high-pressure standby the DDP losses are around one tenth.

Data was only available for idle and full flow for both the commercial products. The behaviour at intermediate displacement may deviate substantially from a straight line joining the two ends. For instance, the losses at 10% flow may be expected to be lower than the standby losses. Under high-pressure standby conditions in an open-circuit swashplate pump, there is no flow of flushing oil to keep the temperature in the case under control, so the temperature in the pump may be much higher than in the rest of the fluid circuit, causing increased leakage than if there is a small amount of flow through the machine.

Without more data on the losses of swashplate pumps at intermediate displacement, it is impossible to give a definitive statement of the magnitude of loss reduction possible with the DDP compared with swashplate pump. However the indications are that the DDP should have between one half and one fifth of the losses of an equivalent swashplate pump, across a wide range of displacements, speeds and pressures.

It should be noted that this is a component-level comparison between open-circuit swashplate pumps and the open circuit DDP. When applied to vehicle transmission systems, the DDP requires external valves to achieve four-quadrant control (Section 5), which must be sized carefully such that the pressure-drop in these valves does not significantly reduce system efficiency. Balancing this, swashplate pumps are typically used in the closed-circuit configuration for propel systems, causing extra losses from the charge pump. The comparison above is valid for propel systems, as long as these extra system losses are equivalent in both cases.

The comparison above is based on the assumption that the efficiency of DDPs, sized to be suitable for mainstream mobile applications (e.g. 40cc to 200cc), is similar to that of the C2 DDP of 12cc displacement. As this thesis was being finalised (November 2006), efficiency results were obtained by Sauer Danfoss for a new DDP (the C8), created by AIP. This was of 96cc displacement, comprising 12 cylinders of 8cc each. This machine displayed a peak efficiency of 96.5%, an encouraging sign that the above comparison with swashplate pumps is substantially valid, at the scale of mainstream mobile hydraulic applications.

2.10 Conclusions

The objectives set out at the beginning of this chapter have been achieved. The efficiency of prototype DDPs has been experimentally investigated. A mathematical model of the energy losses in DDPs has been presented, and validated with the experimental results. The model has been used to compare the DDP with the swashplate pump, showing a potential for dramatic reduction of energy losses.

The most important conclusions of this chapter are:

- The standard definition of 'volumetric' and 'mechanical' efficiency are not appropriate for DDPs, because a significant fraction of the compressibility energy is recovered after each piston stroke. A more appropriate definition has been proposed in which losses are more accurately apportioned to mechanical and volumetric sources, but which relies upon an estimate for the effect of compressibility, derived from geometrical measurement of the working chamber dead volume.
- Experimental results have been presented of a DDP which attained 97% (peak) overall efficiency. The machine was particularly efficient at medium speeds and high pressures.
- Existing loss models have been shown to be inappropriate for the DDP.
- A semi-empirical model has been proposed and compared with experimental data. The assumptions made in the model have been validated with experimental results. Methods to identify the parameters from experiments have been described. The model can be easily applied to forward- and backward-facing simulations of energy flow in hydraulic systems.
- Compared to the swashplate pump frequently used in hydrostatic transmissions, the potential has been shown for a reduction in losses to between one half and one fifth, depending on the operating point.

3 Development of valves for the DDPM

3.1 The DDPM concept

The Digital Displacement Pump/Motor (DDPM) was invented by Win Rampen and Stephen Salter, as disclosed in patent EP0494236. This describes a positive displacement piston pump/motor, having a high-pressure and a low-pressure port. Each working chamber is commutated by two solenoid-actuated digital poppet valves:

- a **low pressure valve** (LPV), which controls the connection of the working chamber to a low-pressure port;
- a **high pressure valve** (HPV), which controls the connection of the working chamber to a high-pressure port.

At any one time, each of the working chambers of the machine can be at one of four states or strokes, each of which can last for up to half a revolution:

Stroke	Chamber state	LPV state	HPV state
Intake	Expanding	Open	Shut
Exhaust	Contracting	Open	Shut
Motor	Expanding	Shut	Open
Pump	Contracting	Shut	Open

A repeating combination of these states is termed a *cycle*:

Idle cycle: repeating intake and exhaust strokes.

Full motoring cycle: motoring stroke followed by an exhaust stroke

Full pumping cycle: a full pumping stroke followed by an intake stroke.

Patent EP1537333 by Stein, Caldwell and Rampen extends this by adding partial strokes and algorithms to control them:

A *partial motoring cycle* consists of a motoring stroke transitioning to an intake stroke part way through the stroke, followed by an exhaust stroke.

A *partial pumping cycle* consists of an exhaust stroke transitioning to a pumping stroke at some time part way through the stroke, followed by an intake stroke.

The figure below shows the path taken by the fluid during the idle, pumping and motoring cycle.

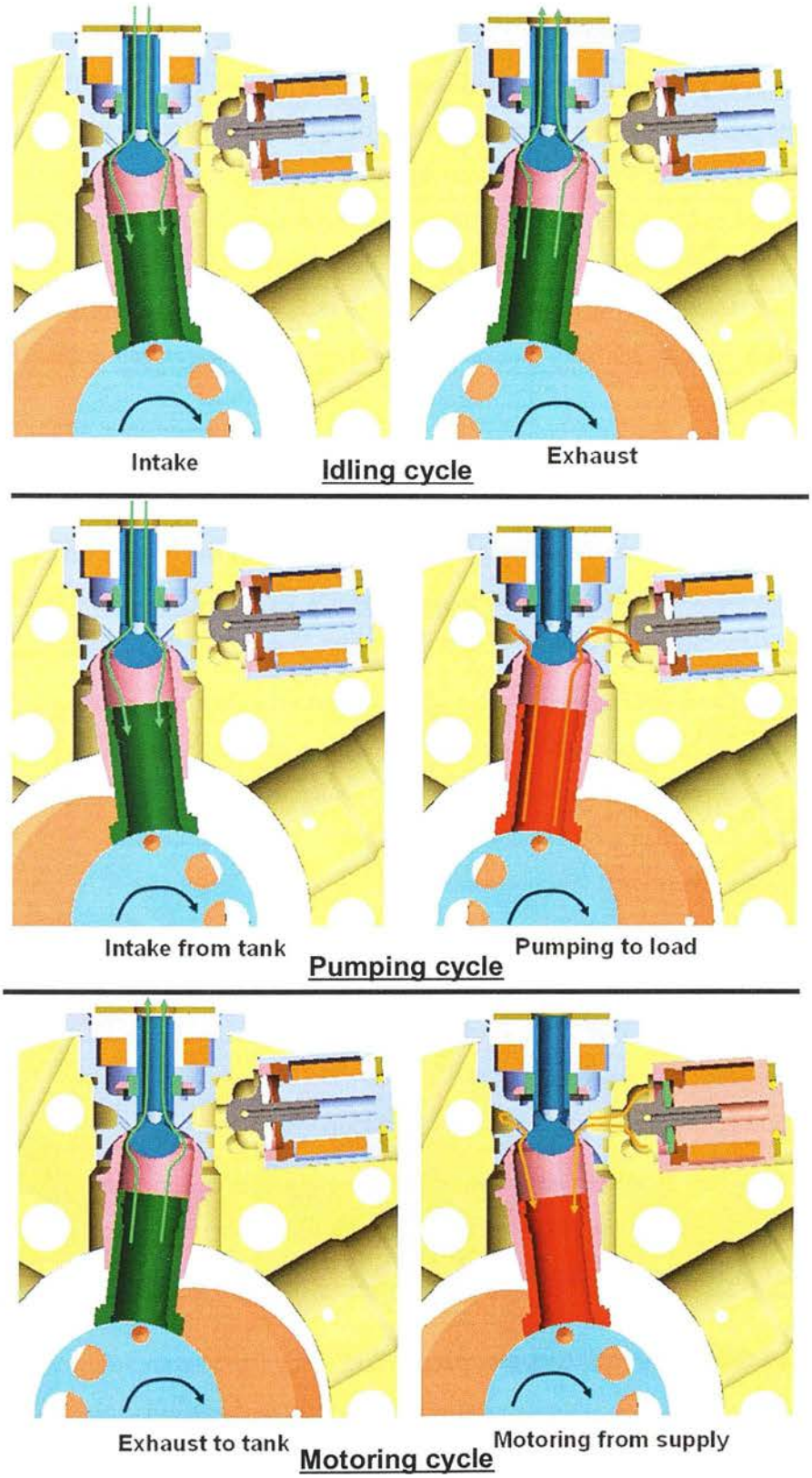


Figure 3-1: Basic DDPM operating cycles showing fluid flow paths

3.1.1 Type of poppet valve

Broadly, poppet valves can be either direct acting or piloted. For the DDPM, the choice between the two is dictated by the extreme speed of operation required: when connected to an industrial diesel engine with typical operating speed of 2500rpm, one complete revolution of the machine takes 24ms. The transition time of the valves must be only a small fraction of this to avoid pressure drops caused by flow moving past half-shut valves – a similar requirement is made of a MOSFET (field effect transistor) used in a switched-mode electrical power supply, which must switch very quickly to avoid resistive losses, which cause over-heating and inefficiency.

Direct acting valves are actuated directly by the force of a solenoid. Solenoids are capable of exerting small forces compared to the larger hydrostatic forces keeping a poppet on its seat, so to operate at high pressure, the active pole area of a solenoid must be very much bigger than the effective area of the poppet. In practice this means that direct acting solenoids can have either large flow-rates or be capable of opening against high pressure, but not both.

Piloted valves use a small solenoid valve which controls the flow to a piston; when the pilot actuates, this piston moves the main poppet by changing the force balance, using a differential area principle. A small solenoid can open a large poppet against pressure, but a piloted valve is inevitably slower than a direct-acting valve, because the operation involves a sequence of events, each with associated time constants.

The valves in a DDP only ever need to be switched when the differential pressure is very low, so the solenoid always has sufficient force to actuate the poppet. However the motoring cycle involves opening the HPV at TDC, at which point the HPV is acted on by the pressure in the high-pressure port. The speed constraint rules out using piloted valves, which are capable of opening against pressure with an acceptably large flow area.

The DDPM patent discloses a method of operating a motor commutated by direct-acting solenoid valves whereby the HPV solenoid does not need to be strong enough to overcome the hydrostatic force on the poppet.

3.1.2 The DDPM motoring cycle

The figures below show the required valve motions and associated valve control signals for the DDPM motoring cycle disclosed by Rampen and Salter, and section views of the state of the machine at important points in the cycle. The machine under consideration has normally-open, solenoid-closed LPV and a normally-closed, solenoid-opened HPV.

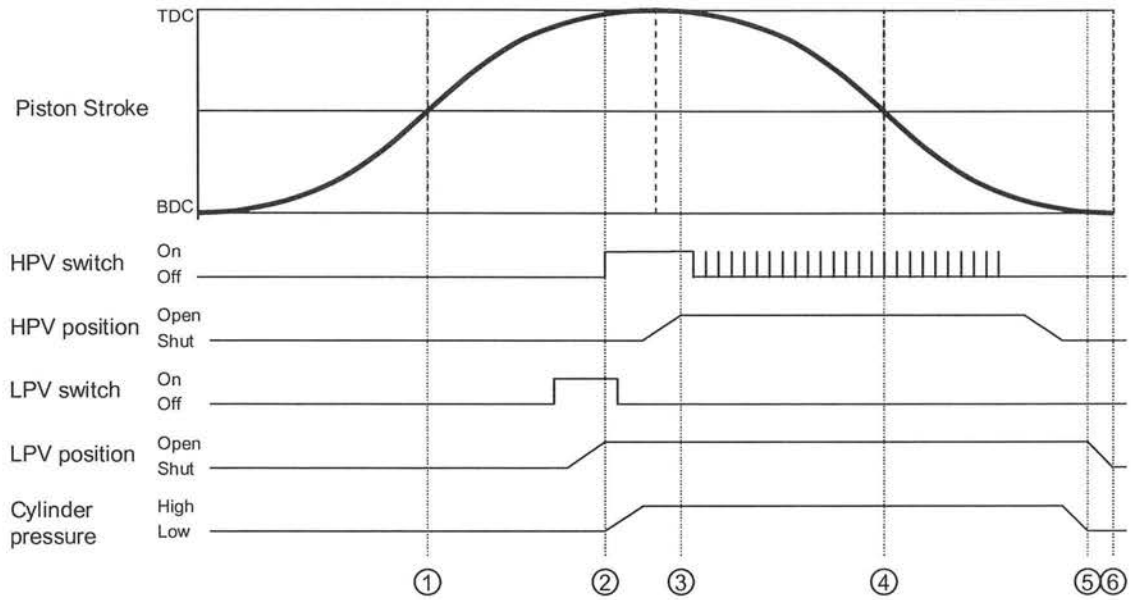


Figure 3-2: Overall timing diagram for the motoring cycle with NOSC LPVs and NCSO HPVs as defined below

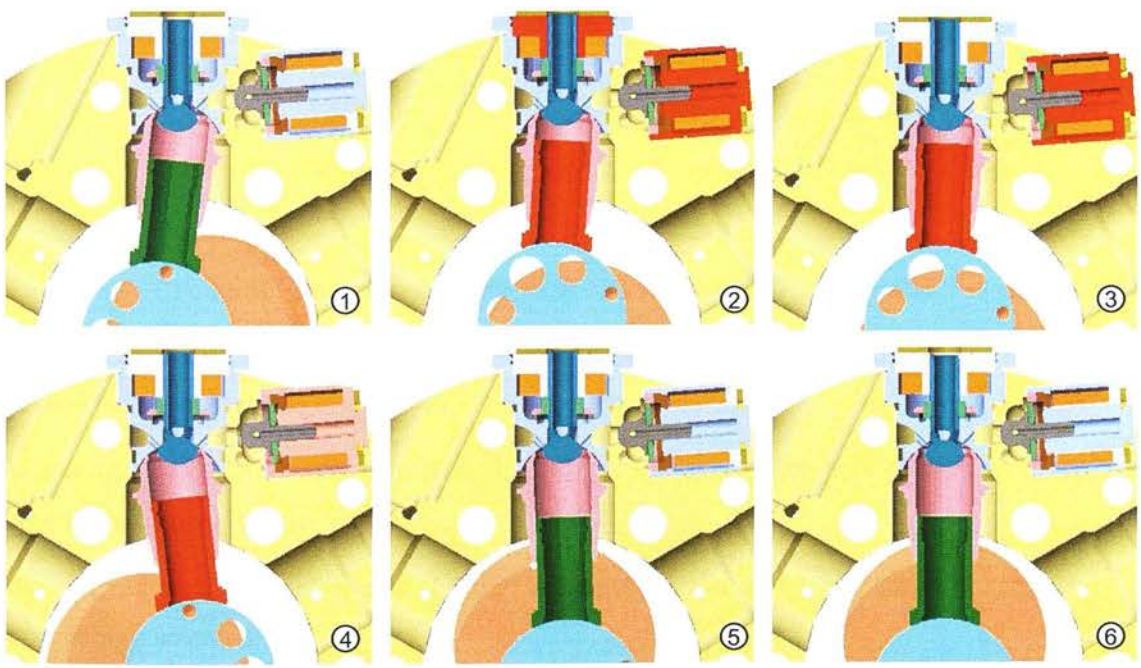


Figure 3-3: Machine states during the DDPM motoring cycle (Coils: blue=off, red=on, pink=pulsed; Piston: red=high pressure, green=low pressure)

Referring to the numbers in the above figures, the DDPM motoring cycle is described in terms of the following significant parts of the cycle:

1. During the exhaust stroke, the HPV is shut and the LPV is open. Fluid from the chamber flows into the low-pressure port.

2. Just before TDC, the LPV coil is energised and the LPV shuts. As the chamber continues to contract, the pressure in the closed volume rises. The HPV coil is energised.

3. When the chamber pressure equals the pressure at the high-pressure port, the hydrostatic force on the HPV poppet reduces to zero and the HPV solenoid force becomes sufficient to open the HPV poppet. Once it is fully open, the current in the HPV coil is reduced to latch it in this state by pulsing the voltage across the coil.

4. After TDC, the chamber expands and does work against the eccentric. High-pressure fluid flows into the chamber through the open HPV during the motoring stroke.

5. Some time before BDC, the HPV latching current is switched off and the HPV closes. The chamber is closed but still expanding, causing chamber pressure to fall.

6. When chamber pressure equals the pressure in the low-pressure port, the LPV is opened by a spring and latched by a permanent magnet. The exhaust stroke begins.

In the partial motoring stroke, event 5 above happens part way through the motoring cycle, so that only a fraction of a full chamber volume is absorbed from the high-pressure port.

3.1.3 Choice of valve type for the HPV and LPV

There are a number of configurations possible of the DDPM, depending on the type of valve chosen for the LPV and HPV. These are shown graphically below.

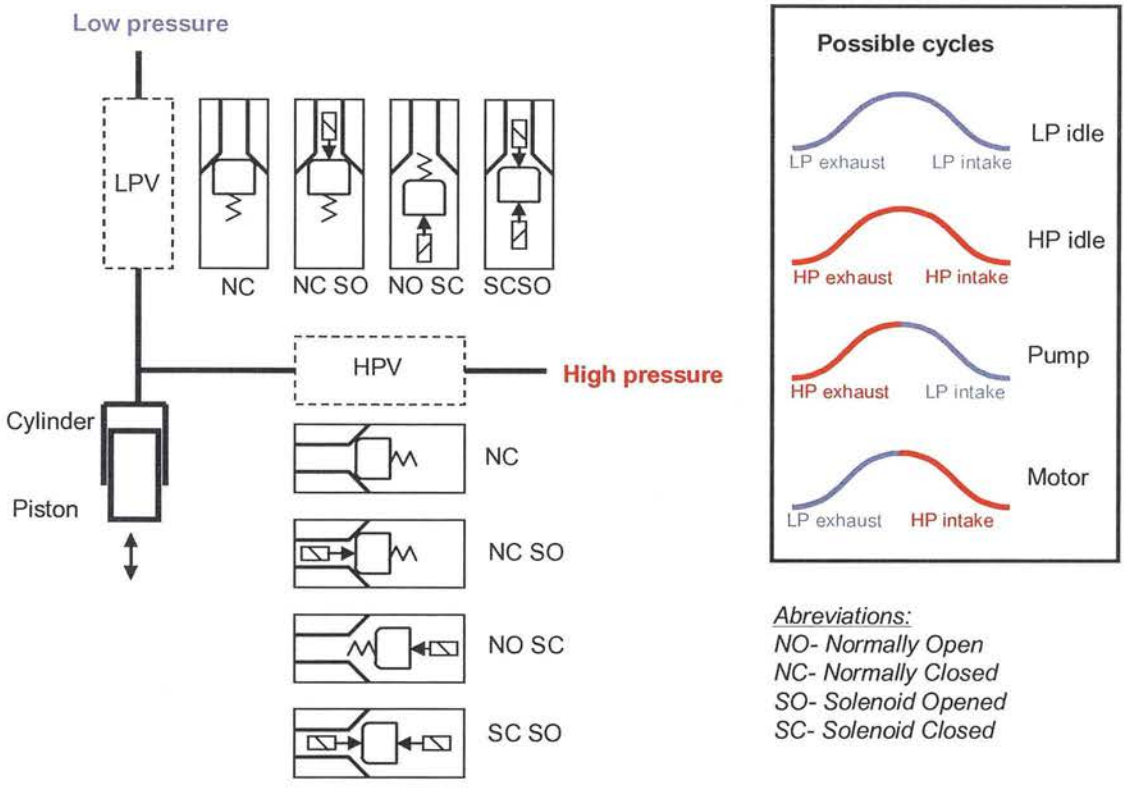


Figure 3-4: Choice of basic valve type for each of LPV and HPV, and possible cycles

In the figure above, the LPV and the HPV can each be of four possible types, defined in terms of how the valve is closed and opened.

Normally closed (NC) – the poppet is closed by a spring or other passive element (e.g. a permanent magnet latch).

Normally closed, solenoid opened (NCSO) – the poppet is closed by a passive element and opened by a solenoid.

Normally open, solenoid closed (NOSC) – the poppet is opened by a passive element and closed by a solenoid.

Solenoid open, solenoid closed (SOSC) – the poppet is opened and closed by two separate solenoids.

The resulting machine can operate in some of the following cycles, one of which may be a default.

LP idle – the chamber alternately contracts and expands at low pressure

HP idle - the chamber alternately contracts and expands at high pressure

Pumping cycle – the chamber expands at low pressure and contracts at high pressure.

Motoring cycle – the chamber contracts at low pressure and expands at high pressure.

In the table below, the functionality of the resulting machine is given for each combination of valve type.

LPV / HPV	NC	NC SO	NO SC	SC SO
NC	Classical check-ball pump. Pumping (default).	DDP. LP idle or pump (default).	DDP as described by Salter & Rampen. LP idle (default) or pump.	DDP. LP Idle or pump. (Default not defined)
NC SO	DDP. HP idle, pump (default).	DDPM. LP idle, HP idle, pump (default), motor.	DDPM as described by Salter & Rampen. LP idle (default), HP idle, pump, motor	DDPM. LP idle, HP idle, pump, motor (Default not defined).
NO SC	DDP. HP idle (default), pump.	DDPM. LP idle, HP idle (default), pump, motor.	Actively enabled DDP. LP idle, pump (defaults to free-flow HP to LP)	Actively enabled/disabled DDP. LP idle, HP idle, pump (default not defined, could be free-flow)
SC SO	Actively enabled/disabled DDP. HP idle or pump (default not defined)	DDPM. LP idle, HP idle, pump, motor (default not defined)	DDPM. LP idle, HP idle, pump, motor (default not defined; could be free-flow)	Actively enabled/disabled DDPM. LP idle, HP idle, pump, motor (default not defined, could be free-flow)

Figure 3-5: Table of DD machine functionality by valve type

In addition to the cycles described, some of the variants are capable of a further mode, in which both the LPV and HPV are closed and the working chamber is at a lower pressure than the low-pressure port. This may have the advantage of reducing losses, as valve breathing is eliminated. It may also cause problems, perhaps drying out the piston pads and causing wear or causing foaming of the oil, or damaging parts due to cavitation erosion. This mode of operation is not considered here but deserves future investigation.

To narrow down the choice of valve configuration, it is important to consider the default mode of operation i.e. what the machine does if the electronic controller takes no action (or has failed).

HP idle is not considered to be a useful cycle and may even be damaging. This cycle exposes the working chamber to the full pressure of the high-pressure port all of the time; this would cause continual losses due to leakage and shear, as well as extra wear and heat generation in the highly loaded bearings such as the piston pad. Those variants which default to the *HP idle* mode are therefore excluded.

The SCSO variants have a lack of definition about how the machine behaves in the default mode which makes analysis of the effects of failure modes difficult - it may be that the default mode is different depending on speed, pressure or some other parameter. Also they require two solenoids and associated coils and electronics for each valve rather than one, increasing cost and complexity. For these reasons they are also excluded.

Also important to consider is whether the machine can get into a state that jeopardises safe operation of other systems to which it is connected. One such condition is "free-flow", the direct connection of the high-pressure port to the low-pressure port. A machine which defaults (or fails) to the free-flow condition may prevent build-up of pressure in the high-pressure port, meaning that working or propel functions of the vehicle may be disabled. In the case of a four-wheel-drive vehicle, the failure in this mode of one of the motors would prevent the other three motors from braking or accelerating the vehicle. For this reason variants which default to the free-flow condition are also excluded.

The excluded variants are shown in the table above with a grey background. This leaves five useful configurations. Of these, the three with HPV=NC provide only pumping functionality. This leaves two configurations which provide DDPM functionality: LPV=NCSO, HPV=NCSO and LPV=NOSC, HPV=NCSO.

For a propel motor, it is crucially important how the motor behaves when the controller has failed or its power is switched off. The DDPM as first described by Salter and Rampen was LPV=NOSC and so defaults to the low-pressure idle mode. In a vehicle propel application this would mean that the vehicle could roll down hill uncontrollably, forcing the designer to fit fail-safe friction brakes capable of halting the vehicle. By contrast a machine with LPV=NCSO defaults to the pumping cycle. This would rapidly retard a runaway vehicle in the event of a power failure by dissipating vehicle kinetic energy in the system relief valve. Brakes may still be needed to prevent the vehicle creeping downhill due to leakage in the motor; but these may be simpler and cheaper if they do not need to be capable of halting the vehicle from full speed, and there is a redundant method of halting the vehicle. This vehicle safety consideration was the main reason that NCSO LPVs were selected for the DDPM propel motor development.

3.1.4 Choice of basic solenoid configuration

The basic design of the solenoid has changed little since Roters' (1945) seminal work and much of it stands as true today as it did then. Roters' contribution was to bring together in one volume:

- the relevant electromagnetic theory, which by that time was well understood and has since changed little;
- a detailed survey of the then-available magnetic materials;
- a "cookbook" of numerical methods to tackle typical solenoid design tasks, from coil design to time-domain transient simulation by iterative hand calculations;
- and an overall classification of the merits of different solenoid designs, much of which was informed by his own experiments.

Roters gives guidelines of the application of the different types of solenoid for optimum weight, using an index number based on the square root of force divided by the travel. For Rampen's (1992) force and travel requirements, the optimum is given as the flat-faced plunger magnet. However Rampen initially rejected this format because of the difficulty of preventing the radial gap of a plunger-style magnet from fouling up with ferromagnetic particles. For this reason, in his initial work on DDPs, Rampen used a design based on a flat-faced armature, in which flux enters and leaves the armature axially. According to Roters this form has the highest possible tractive force, but this force diminishes rapidly as the armature moves away from the pole face.

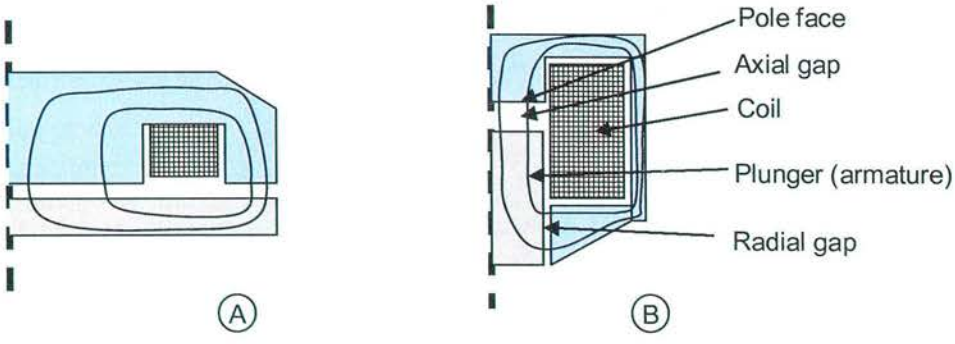


Figure 3-6: Solenoid formats according to Roters (1945) considered by Rampen (1992); (A) flat-faced armature (B) flat-faced plunger

The plunger in the figure above is reported by Roters to be guided by a thin brass tube which extends all the way from the radial gap to the inside of the coil. The tube guides the plunger in the centre of the radial gap; without it, the plunger would always be pulled by the radial flux away from the axis, causing unbalanced radial forces which would generate undesirable

friction. The plunger slides in this tube with just sufficient gap for smooth running. Roters quotes a radial clearance of 0.05mm inside a tube with 0.5mm wall thickness, as being typical for a solenoid with a 38mm diameter plunger. The non-magnetic material in the radial gap adds to the reluctance of the circuit, so it is desirable to minimise the radial thickness of the radial gap.

Towards the end of his studies Rampen conceived of a configuration of the plunger solenoid which removed the concern about fouling. By separating the guidance from the radial gap, he could maintain a larger radial gap without running the risk of the armature jamming against one side. This is the configuration used for all the DDPM valve solenoids described here.

Although the basic configuration of solenoids has changed little since Roters was writing, technology advances have changed how solenoids are designed and applied:

- the development of rare-earth magnets has allowed latching coils to be eliminated from some applications (Rampen 1992);
- high-temperature superconductive wire has been used to create high field strengths for the most demanding applications such as high-energy particle physics, although this has yet to make much impact in the commercial engineering world. (Myatt et al. 1994);
- the development of power electronics capable of high-frequency switching operation now allows solenoids to be proportionally modulated without excessive power losses (Kajima et al..1992);
- microcontrollers are inexpensive enough to be embedded into applications, allowing digital solenoids to be timed precisely in common rail fuel injection (Ricco et al. 2004), and proportional solenoids used in diesel engine exhaust gas recirculation to be linearised, by inferring their position from the impedance they present to a switching circuit (Rahman 1996);
- most important, the power of the modern digital computer allows every engineer engaged in solenoid design access to finite element analysis tools with which to calculate flux distribution, and therefore force and inductance (Meeker 2006) .

3.1.5 Review of basic solenoid characteristics

At this stage it is useful to review the fundamentals of operation of a plunger-type solenoid. A simple one-dimensional model of the NCSO HPV solenoid is shown below.

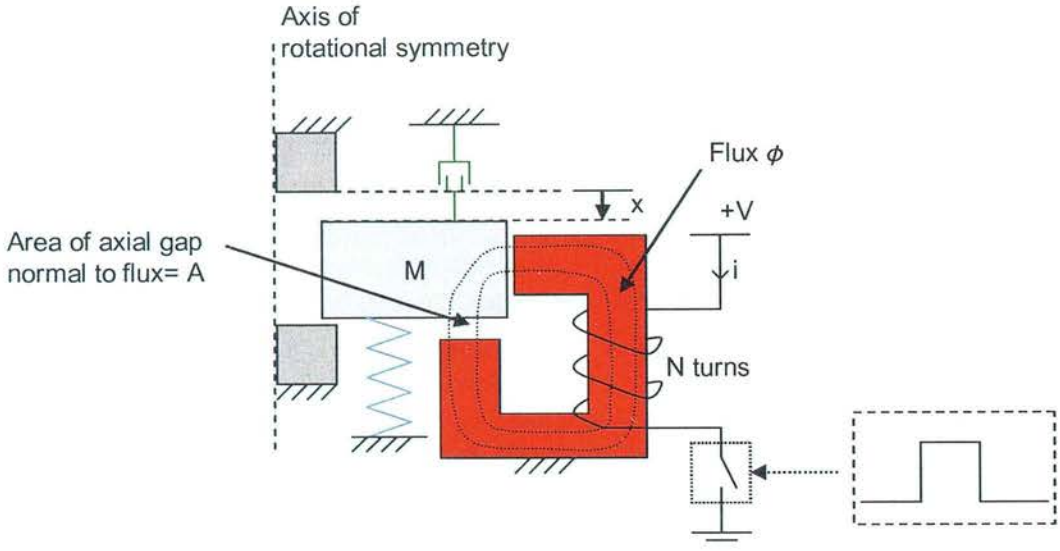


Figure 3-7: Ideal one-dimensional model of solenoid valve motion

The poppet is represented by a mass M free to move between two hard end-stops. A return spring forces the poppet towards the closed condition (x =0 above). The motion of the poppet is resisted by the viscous damping of the surrounding oil; the damping constant may be variable dependent on poppet position due to squeeze-film effects at the end-stops. The poppet armature is part of a magnetic circuit, and is actuated by the force created by flux in the axial air gap. At low flux density, the most significant reluctances in the circuit are the axial and radial air gaps, but as flux density in the steel parts of the circuit approaches the saturation point, significant extra reluctance is introduced in the steel itself.

A coil of wire of N turns carrying a current of i amps provides the magnetomotive force Ni to create the flux ϕ in the magnetic circuit. The axial force on the armature is generated in the axial gap with area A; here the flux density $B= \phi/A$. In the electrical domain, this coil has a resistance which is a function of the length and diameter of wire and the wire temperature, and an inductance which is a function of the number of turns of the coil and the total air gap in the circuit, and hence armature position.

Solenoid actuators are inherently non-linear. The force on the armature increases with the second power of flux density (Roters 1945).

$$F = \frac{B^2 A}{2 \cdot \mu_0} \tag{40}$$

Where F is the attractive force (N)

B is the flux density (T)

A is the area of the pole normal to the flux (m^2)

μ_o is the permeability of free space = $4\pi \cdot 10^{-7}$

The above equation indicates that to achieve maximum force, the flux in the axial gap should be concentrated into the smallest possible area; as the area reduces, flux density increases according to the inverse square of the area, and the net force increases in inverse proportion to the area.

When the armature moves towards the pole, the axial air gap reduces in length, reducing the reluctance of the magnetic circuit. With a constant ampere-turns of magnetomotive force, this causes the flux to increase.

Assuming linear behaviour of the magnetic material, flux density is proportional to ampere-turns, and for an ideal plunger-type solenoid there exists a simple formula for the attractive force between the pole and armature (Roters, 1945; Chladny et al. 2005):

$$F = \frac{\beta \cdot i^2}{(k - x)^2} \tag{41}$$

where β is a constant relating to the number of turns in the coil and the area of the flux paths, while k relates to the length of the radial flux path. This behaviour is shown graphically below.

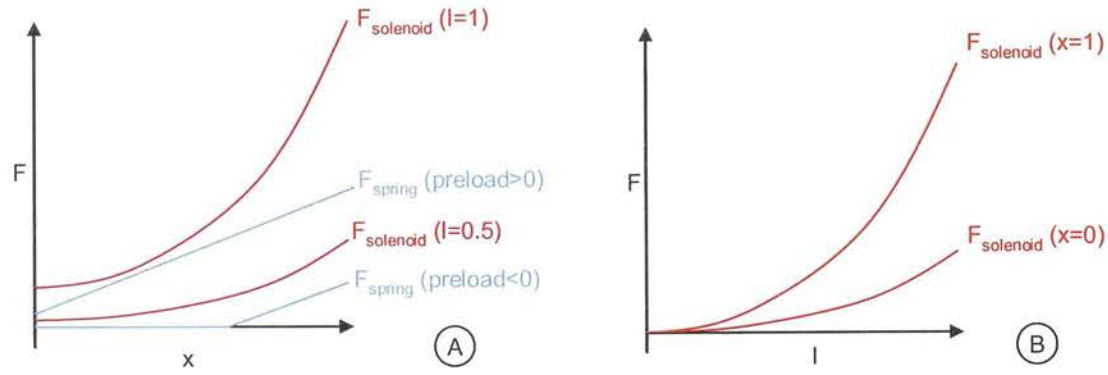


Figure 3-8: Ideal force vs. displacement (A) and force vs. current (B) characteristics

(A) shows the qualitative relationship between force and armature position with two values of coil current. Also shown are two possible lines of return spring force vs. armature position, in the cases of negative and positive spring preload.

(B) shows the qualitative relationship between force and coil current at the two extremes of position.

The simple solenoid model in eq. 41 above is only valid if the magnetic materials have a constant permeability, and if all the magnetic flux flows through the axial gap normal to the axis of the poppet. However, due to the non-linear effects of saturation and leakage, this is not true of practical solenoids designed with highest force as the objective. Finite element methods are typically used in this case.

3.2 Early DDPM development

3.2.1 Background

In 1999 Sauer Danfoss expressed an interest in the development of a DDPM capable of propelling a demonstration vehicle. By this time Rampen had created a single-cylinder DDPM which was directly connected to an induction motor with a nominal speed of 1500rpm. This was capable of moving a single-acting lifting cylinder in both directions, the energy from lowering the load being regenerated back into the shaft. This was demonstrated by lifting a chair with a modified engine hoist with a joystick demand.

As there was only one chamber, the flow produced was strongly pulsating and this could be felt by the payload. It was obvious that a practical motor for propelling a vehicle would need a number of cylinders to reduce the pulsation level. A 6-cylinder radial piston DDP of 12cc/rev had also been created by Rampen, Stein and Almond by this time. As a stepping stone towards a propel motor, the author converted this machine into a multi-cylinder DDPM.

3.2.2 The first-generation valves for the multi-cylinder DDPM

The starting point for the multi-cylinder DDPM was a 12cc/rev, 6 cylinder radial piston DDP power-pack designed by Rampen – with NOSC LPVs and NC HPVs. The general arrangement of this machine is shown below:

Figure 3-9: The 6 cylinder radial piston DDP by Rampen

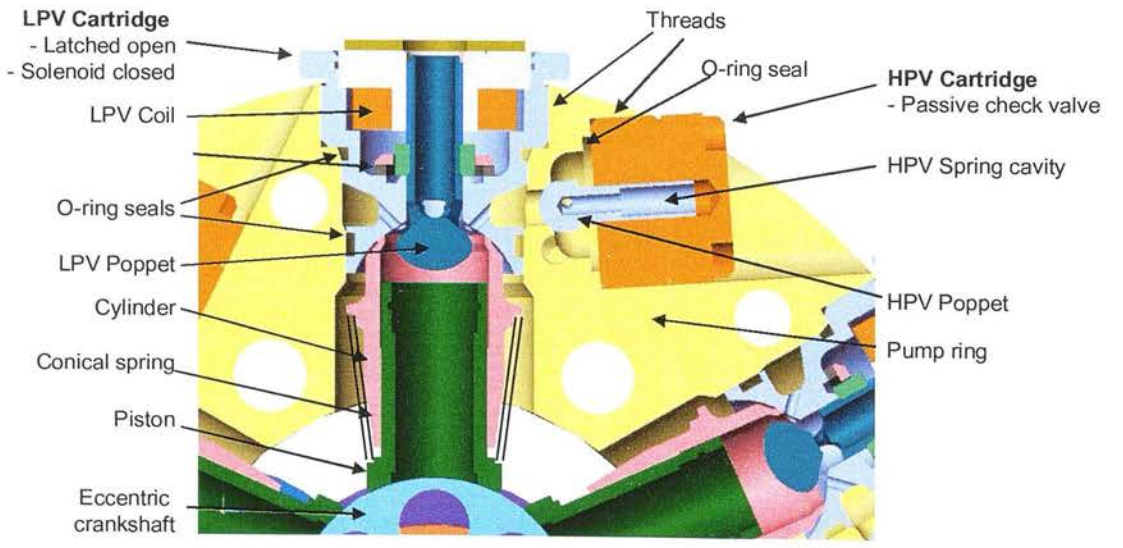


Figure 3-10: Detail of a single DDP pumping element

Rampen had already designed NCSO HPVs for this upgrade, but the NOSC LPVs were unmodified from the original DDP. The DDPM functionality required a return spring to re-open the LPV quickly after de-compression at BDC. The author added a spring to the NOSC LPVs in addition to the existing permanent magnet latch. Electronics were upgraded to provide FET switches for each HPV coil.

The figures below shows the 'baseline' NCSO HPV designed by Rampen, the NOSC LPV by Rampen (modified by the author with a return spring), and both valves in the solid model assembly.

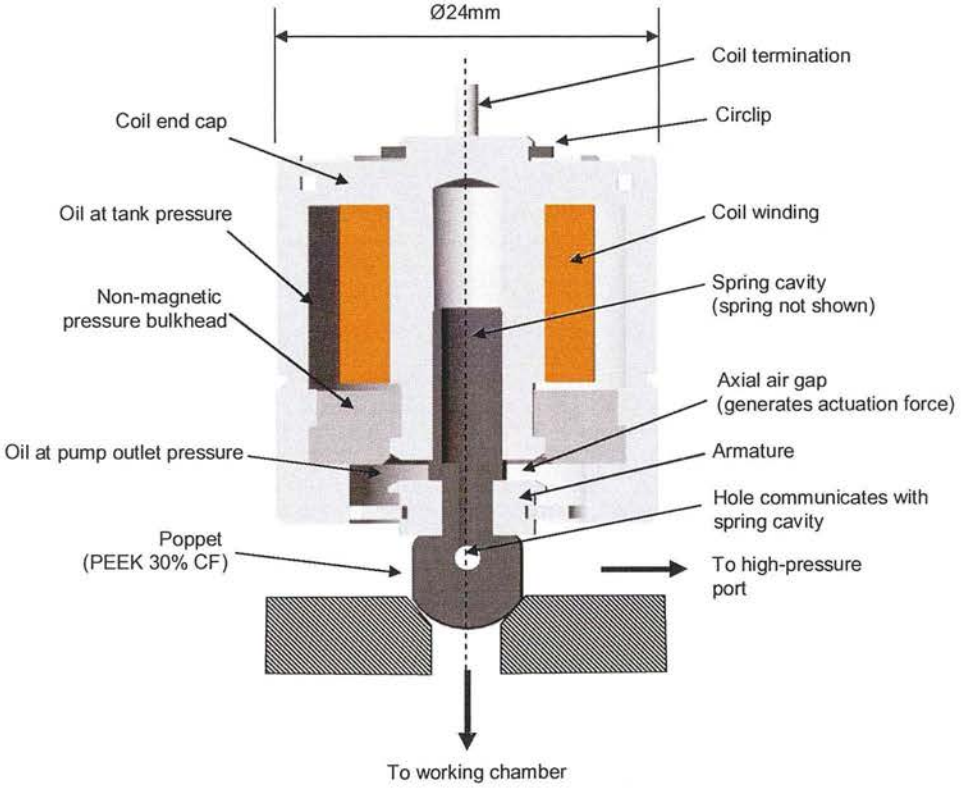


Figure 3-11: Section of the baseline NCSO HPV design, designed by Rampen

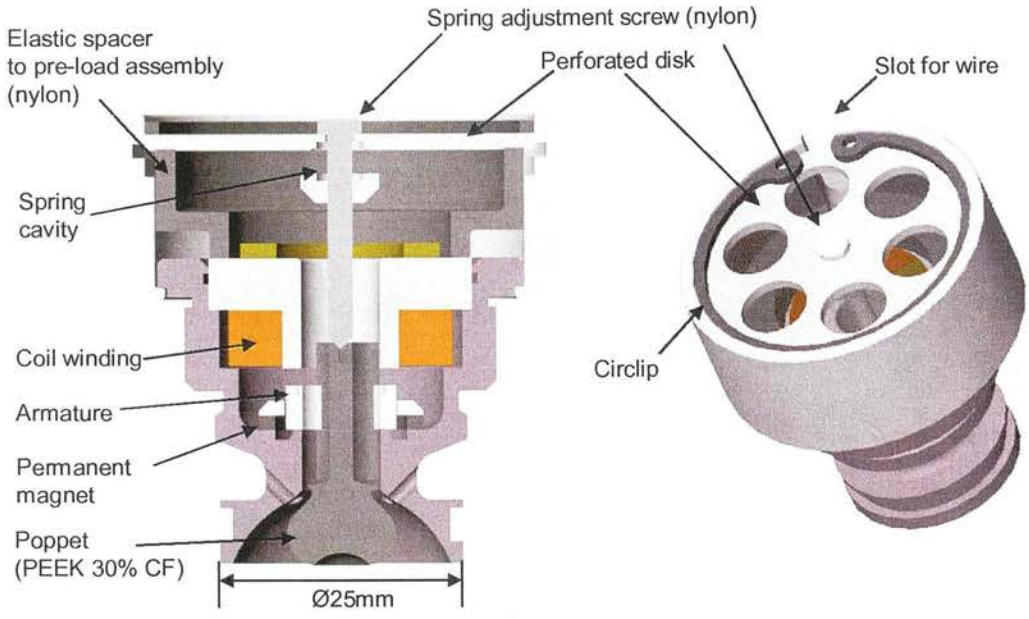


Figure 3-12: Detail of the first generation NOSC LPV by Rampen, with adjustable return spring arrangement by the author

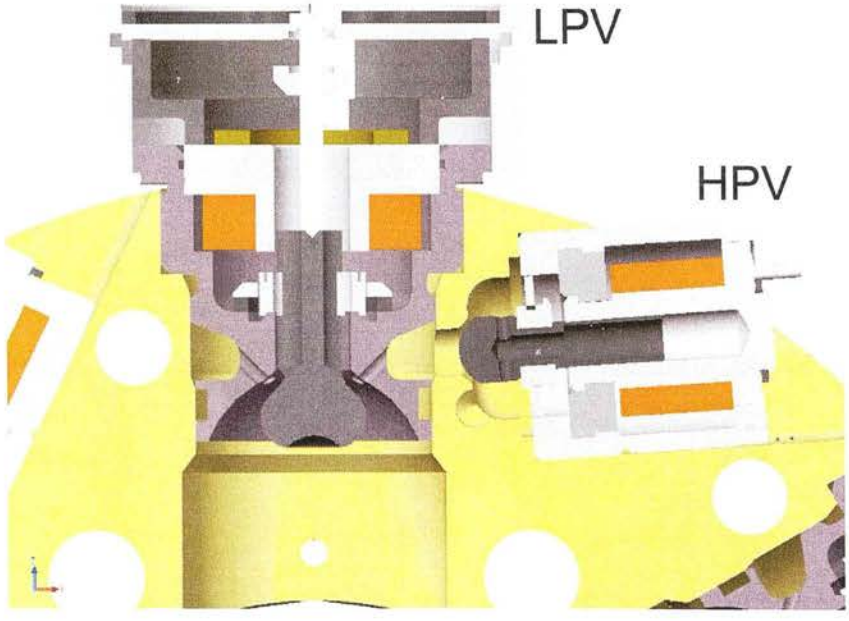


Figure 3-13 The first generation DDPM valves in the solid model assembly

3.2.3 Tests with non-contact position sensors

During the development of the DDPM, it was necessary to be able to measure the positions of the LPV and HPV poppets in real time, so that the time of the actual closure of the valves could be measured.

To this end, a non-contact position measurement method was used, whereby the flux from a small permanent magnet connected to the poppet was sensed by a linear Hall Effect chip (Melexis MLX 90251). As the poppet moved, the sensor produced a voltage proportional to the flux through the sensor chip; this voltage was measured with an oscilloscope triggered by the once-per-rev shaft position sensor. The photo below shows the initial development on the bench of this technique, for the LPV:

Figure 3-14: Non-contact measurement of LPV poppet position with linear Hall effect sensor. Later the arm was replaced with a bracket, allowing measurements in a working machine.

Initial tests showed that because the sensor was not perfectly linear, the sensor signal during the transit was not linearly related to the poppet position. Another drawback was that moving mass of the poppet was increased by about 10% by the permanent magnet sensing target, so the instrumented valves could be expected to be somewhat slower that the non-instrumented valves. However the technique did provide useful data of the transit time of the poppet from one endstop to the other. It is the magnitude and variation of this total transit time which is of most interest when characterising the valves.

A single LPV and HPV were modified to allow non-contact position sensing of the valves during operation of one cylinder of the 6-cylinder DDPM. This was straightforward for the LPV, but for the HPV the sensor had to work through a non-magnetic pressure bulkhead, as shown below.

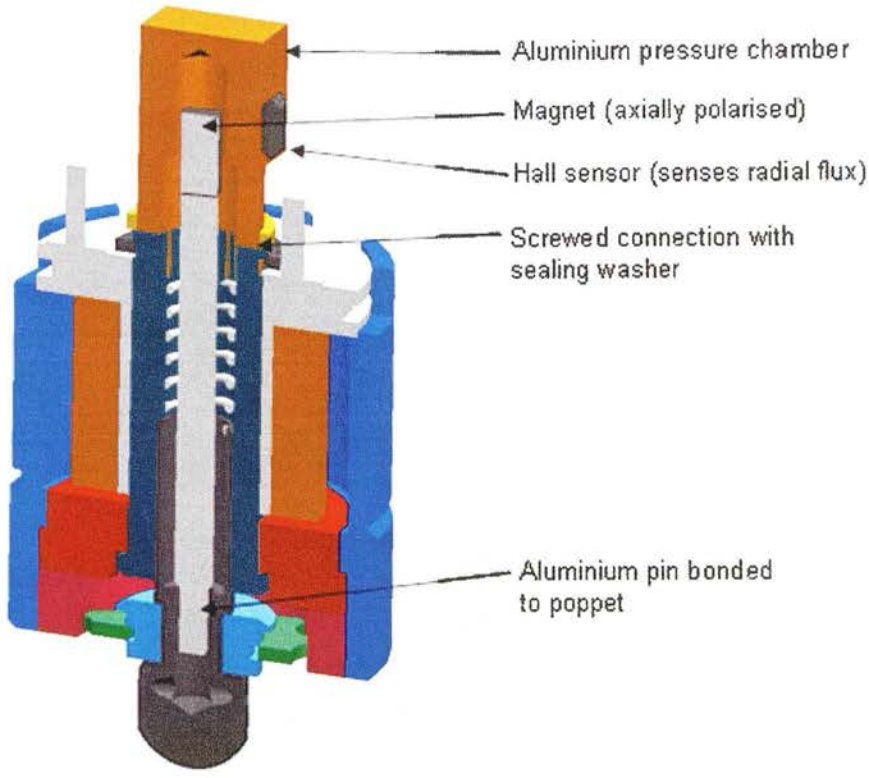


Figure 3-15: The non-contact position sensing arrangement for HPV, complicated by the need to sense through a pressure bulkhead.

For early tests the response of the valves was investigated on the bench, outside of the DDPM – this allowed the basic electromechanical dynamics of the valves to be studied without flow or pressure forces on the poppet. The scope trace results shown below are from a later valve design, but using the same position measurement technique. Annotations show the definition of timing measurement used to characterise the valve response time.

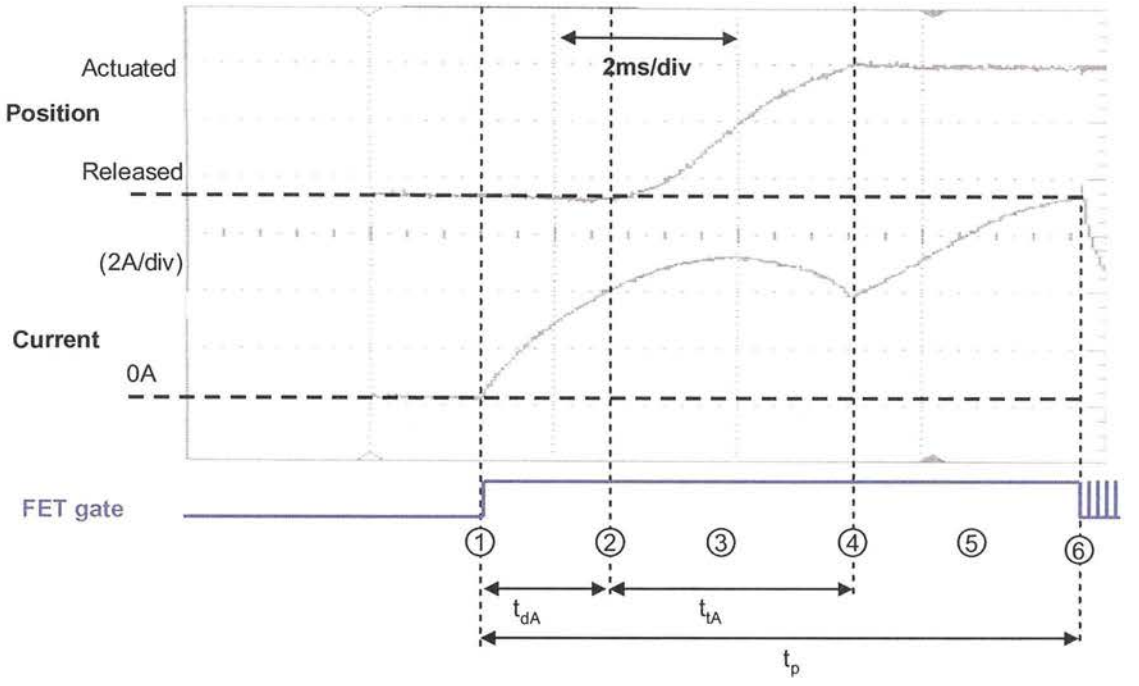


Figure 3-16: Definition of timing measurement for valve characterisation; trace from Mk.2 LPV for DDPM propel motor @ 14V in 32cst oil at room temperature.

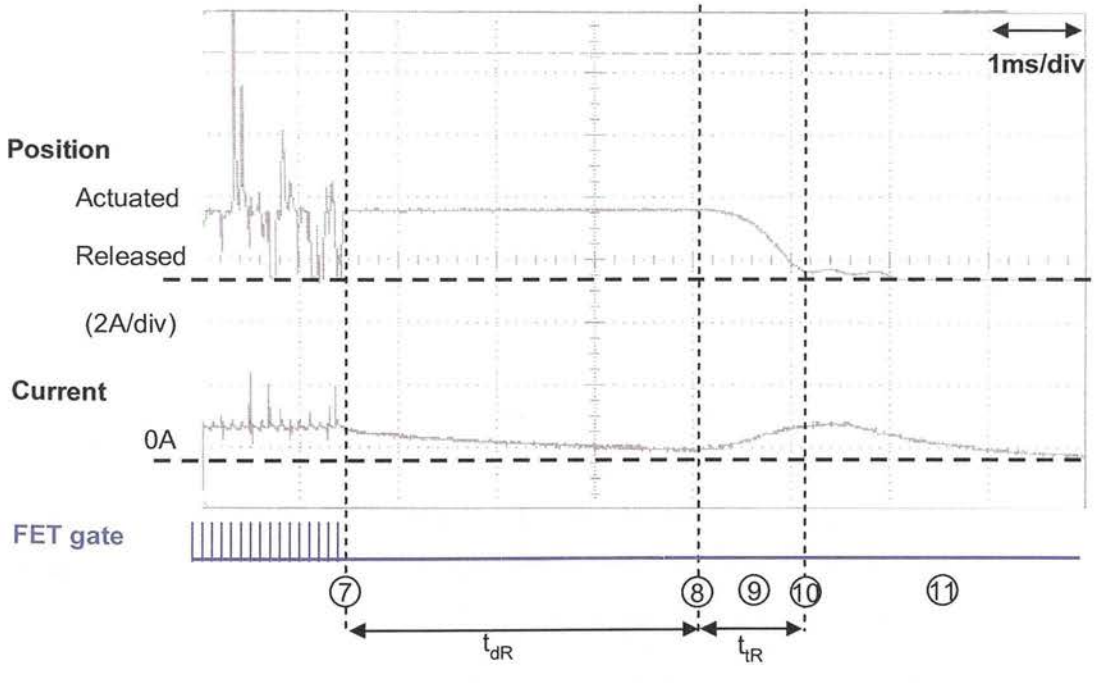


Figure 3-17: Timing parameters for the same valve switching off from pulsed gate signal (22% duty cycle @ 10kHz)

The stages in valve operation are described below with reference to the labels in the two figures above:

1. The FET switches on and current starts to build in the coil. Initially the solenoid force is lower than the spring pre-load, so the armature does not move. While the flux in the magnetic circuit is in the linear portion of the B-H curve (i.e.. below 1.0T with EN1A mild steel), the current rises according to a simple first-order time constant, $\tau=L/R$.

   If the spring pre-load is high, it is possible that at some point the flux exceeds the linear threshold of 1T. The effect will be to reduce the inductance of the coil, thus increasing the gradient of the current rise above what would be predicted by linear theory.

   The time delay between the start of the gate pulse and the start of the actuation of the poppet is designated t_{dA}.

2. At this point, the force from the solenoid exceeds the spring pre-load and the armature starts to accelerate. At around mid-stroke, an equilibrium has been reached between the forces from the solenoid, the return spring and fluid damping. After this point the armature starts to decelerate as the return spring force increases.

3. The armature approaches the pole and the current starts to dip, an effect noted by Roters (1945) and Kawase et al. (1991). The "motional e.m.f." produced by the motion of the armature through the magnetic field acts to resist the current while the armature is in motion.

4. The armature motion is stopped suddenly when the poppet hits the seat. This causes an inflection in the current trace because the motional e.m.f. (which is proportional to armature velocity) suddenly reduces to zero.

   The transit time taken for the valve fully to actuate from the start of motion is designated t_{tA}

5. If the gate pulse to the FET is kept on after the poppet is seated, the current continues to rise but by this time the current is so high and the gap so small that flux density (at least in some parts of the magnetic circuit) exceeds the linear portion of the B-H curve. As the current continues to rise, the incremental inductance falls, and the current quickly reaches the steady-state value of V/R_c.

6. The gate pulse to the FET ends. The current decays as it continues to circulate around the coil via the flywheel diode and its series resistor (see details of the electronic circuit in Section 3.2.5, Figure 3-21). The fixed voltage drop of the diode and the resistive drop of the flywheel resistor have the effect of decreasing the time-constant compared to the rise time, but balancing this is the fact that the inductance with the valve closed is higher than when it was open. If the valve is to be latched with the

solenoid then gate pulses begin at a frequency substantially higher than the reciprocal of the coil time-constant; the pulse frequency was 10kHz in this case.

The duration of the initial gate pulse is termed t_P.

7. The latching period is over and the gate pulses stop. The current starts to decay and with it the latching force.

8. The latching force has decayed to the point that the spring starts to return the poppet towards the default position. The delay time taken for the poppet to start to move after the end of the latching pulses is terms t_{dR}.

9. During the return transit of the valve, the inductance of the solenoid rapidly decreases and the motional e.m.f. causes an increase in current, the reverse process to what happened in 3 above.

10. The armature is fully returned to the default position. The transit time taken to for the poppet to transition from the actuated to the released state is termed t_{tR}

3.2.4 The DDPM development test rig

The system schematic and a photo of the DDPM development test rig is shown below:

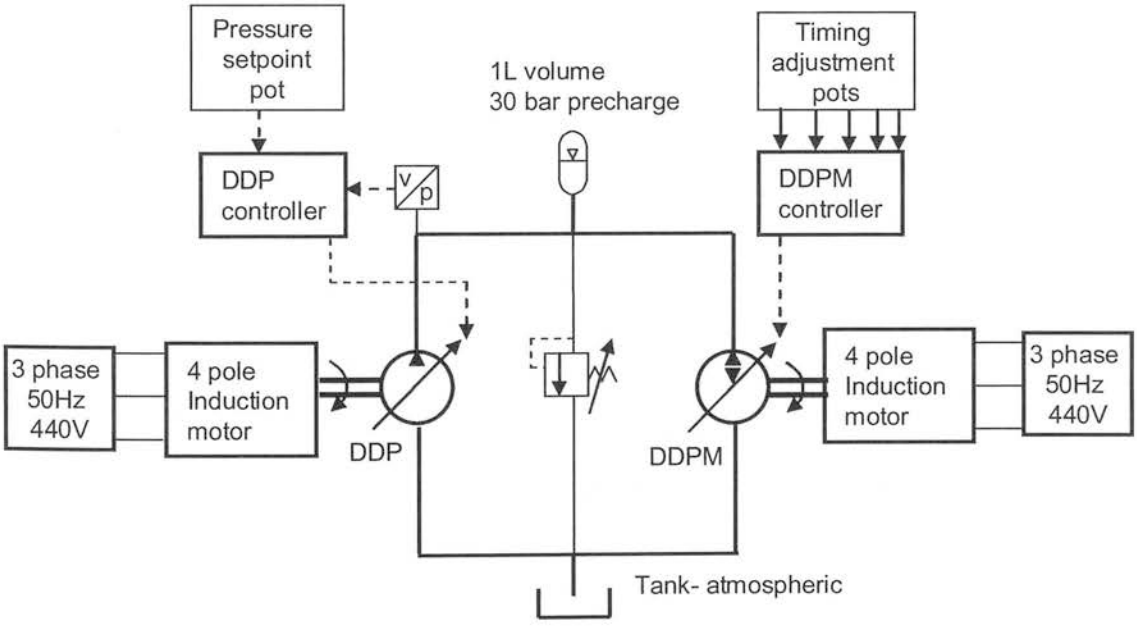


Figure 3-18: System schematic for initial DDPM tests

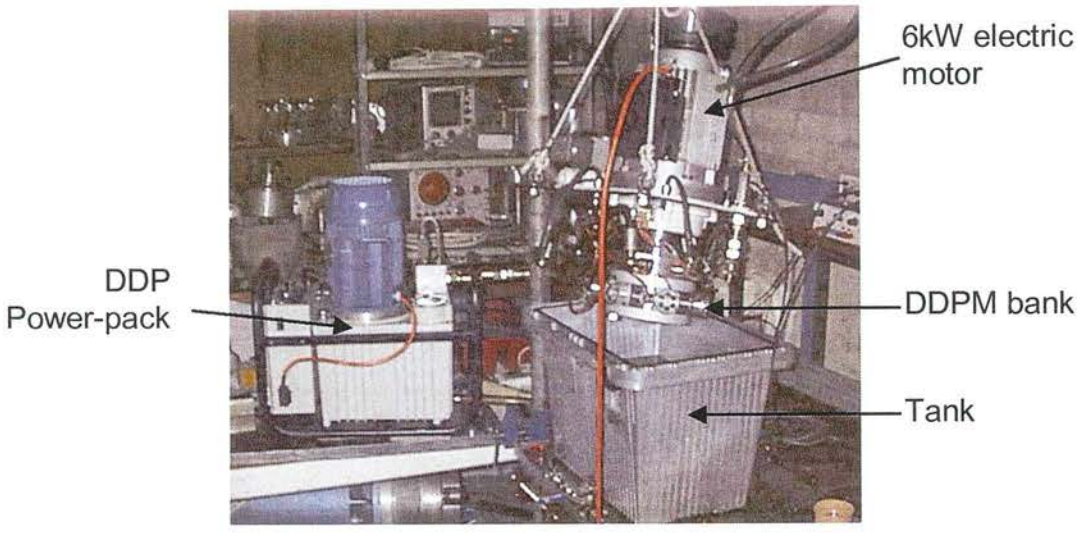


Figure 3-19: The multi-cylinder DDPM, and the DDP used to supply it with fluid

The shaft of the DDPM was connected to a 6kW induction motor running at a nominal 1500rpm. This steady speed simplified the early development of the valves and the control algorithm for two reasons:

- the controller did not need to vary the pulse timing to compensate for shaft speed variations;
- the shaft speed remained fairly constant regardless of whether the machine was pumping, motoring or idling; thus the parameters necessary to achieve a stable motoring cycle could be investigated without the machine speed reacting to the torque being produced.

The pressure supply to the DDPM under test was a 6 cylinder 12cc/rev DDP, operating in pressure control mode, driven at a nominal 1500rpm by a single-phase induction motor. A small gas accumulator smoothed the pressure ripple from this machine to approximately +/- 10 bar. A pressure relief valve was set to relieve at a pressure above the set-point of the DDP pressure control loop. This arrangement simplified the initial testing of the DDPM because the pressure remained fairly constant regardless of whether the DDPM was pumping, motoring or idling.

3.2.5 DDPM software and electronics

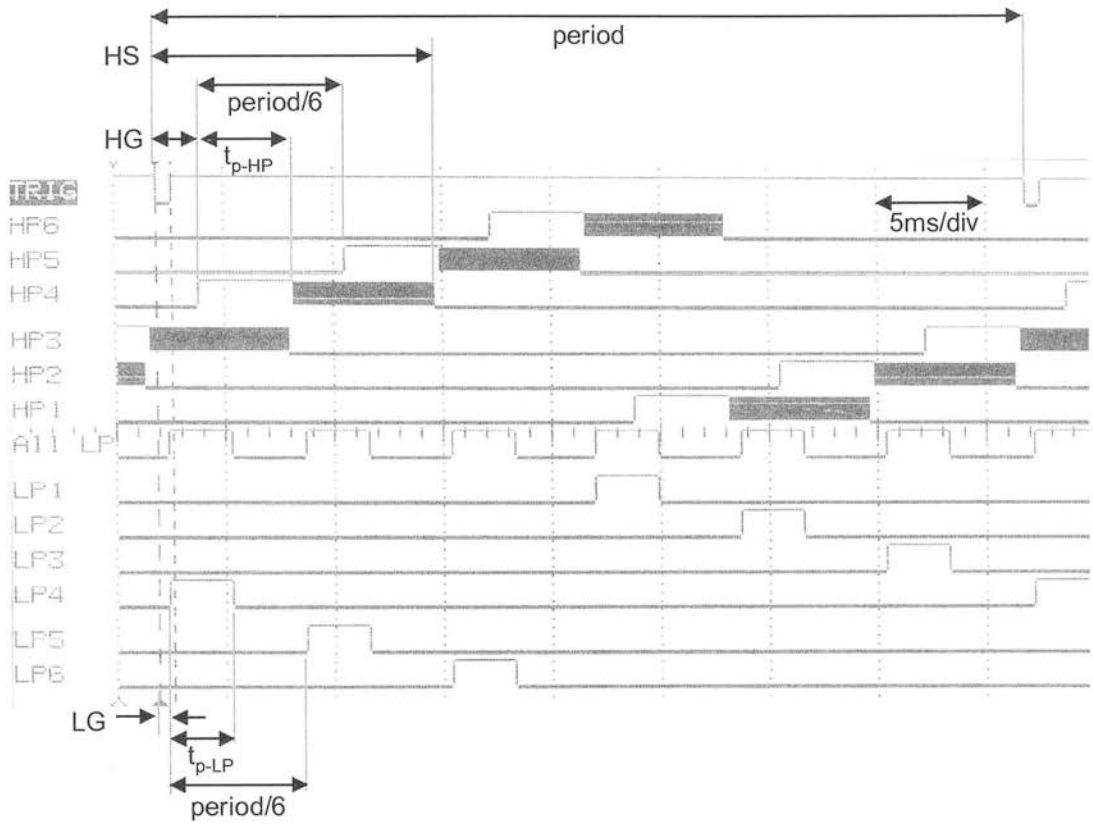


Figure 3-20: FET pulses for the 6 cylinder DDPM motoring at 1500rpm

Jon Almond designed and built the microcontroller board based on an Intel196 16 bit microcontroller. A magnetic inductive sensor sent a pulse (the 'trigger pulse') to this controller, once per revolution, synchronised with BDC of cylinder no. 6. The controller measured the elapsed time between successive edges of the trigger and used this information to determine the period of rotation. Due to each working chamber being separated by 60 degrees, dividing this period by 6 gave the separation of the drive signals for successive valves.

The DDPM valve coils were energised by a FET circuit, the gate signals to which were generated by the microcontroller, as shown in the figure below.

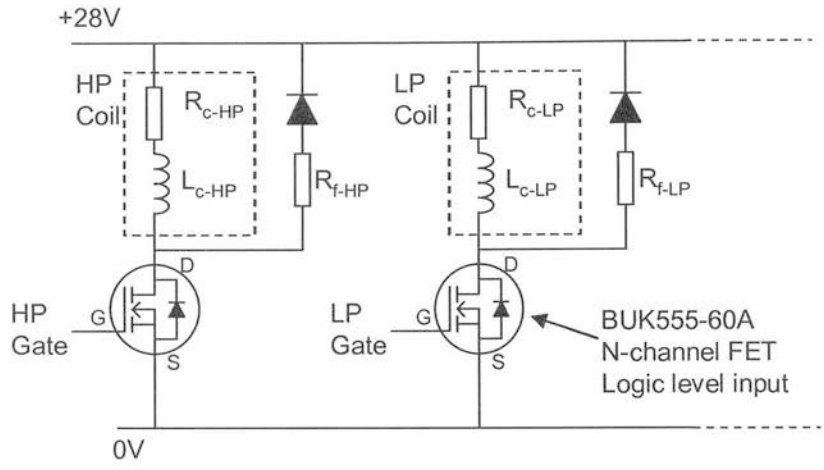


Figure 3-21: Coil drive circuit used in the DDPM

Because of the significant inductance of the valve coils, "flywheel" diodes were connected in parallel to the coils to allow the coil current to re-circulate when the FET drivers were switched off. To speed up the decay of the coil current, resistors were added in series with these diodes ($R_{f\text{-}HP}$ and $R_{f\text{-}LP}$ above); the choice of this series resistance is discussed later.

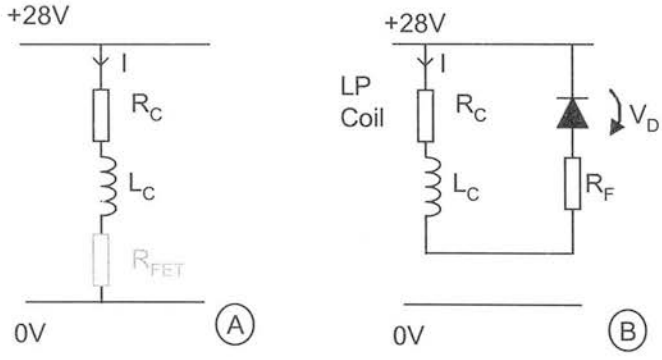


Figure 3-22: Coil drive equivalent circuits with a flywheel resistor: A=FET Gate on; B=FET Gate off

The figure above shows the principle of the flywheel resistors. When the FET is on, the dominant resistance in the circuit is the coil itself (the FET resistance is negligible). When the FET is off, the coil continues to flow in the flywheel diode, through the series resistor. This has the effect that the time constant to switch off the coil $\tau=L/(R_c+R_F)$ is shorter than the time constant to switch on the coil $\tau=L/R_c$.

Each FET was driven by a separate digital output from the controller. To reduce the workload of the controller, intermediate electronics controlled the duration of both LPV and HPV pulses and the duty cycle of the 10kHz pulsed portion of the HP drive signal. With reference to Figure 3-20 above, the timing of the start of the LP pulse (*LG*) and the start (*HG*) and end

(*HS*) of the HP pulse sequence were controlled directly by the microcontroller, but the duration of the pulses ($t_{p\text{-}LP}$ and $t_{p\text{-}HP}$) were controlled by potentiometers setting the period of a monostable oscillator.

3.2.6 Early motoring tests

Initially, tests focussed on achieving a stable motoring cycle. The traces from the position sensors were used to confirm that valve motions were happening at the correct time.

The method used proceeded in the following stages:

1. **Pressurisation**. *LG* was set well in advance of TDC to ensure that the cylinder was pressurised by TDC. Then $t_{p\text{-}LP}$ was reduced until the valve started to become erratic, and increased by 1ms to achieve stable operation. At this point the LPV was closing just before TDC and the machine was pumping very slightly – the system pressure being limited by the relief valve.

2. **HPV opening.** The HPV pulse duty cycle was set to minimum. Then $t_{p\text{-}HP}$ was set to 6ms - definitely long enough to actuate the poppet. Then *HG* was swept earlier and later until the machine started to motor fluid. The limits at which the motoring cycle failed was found, and *HG* was set mid-way between these limits. At this point the HPV was opening some time after the LPV closed, and was closing well before a full motoring stroke had taken place. The DDPM was operating in a partial-stroke motoring cycle.

3. **HPV latching**. *HS* was set to 10ms after *HG*. The HPV pulse duty cycle was increased until the HPV was observed to latch in the open position. Then *HS* was adjusted to confirm that the time of HPV closing was controllable by adjusting *HS*. At this point, varying *HS* controlled the motoring displacement of the machine and therefore flow from the DDP.

4. **De-pressurisation**. *HS* was increased, the effective displacement of the DDPM increased, because a greater fraction of the chamber was motored from the high-pressure port. At a critical value of *HS*, the machine stopped motoring. At this point the machine was doing *HP idle* cycles, with the LPV always closed and the HPV always open. This established a maximum value for *HS* at which *HP idle* cycles are avoided but the displacement was maximised.

Traces were recorded of the valve position during motoring operation, as shown below.

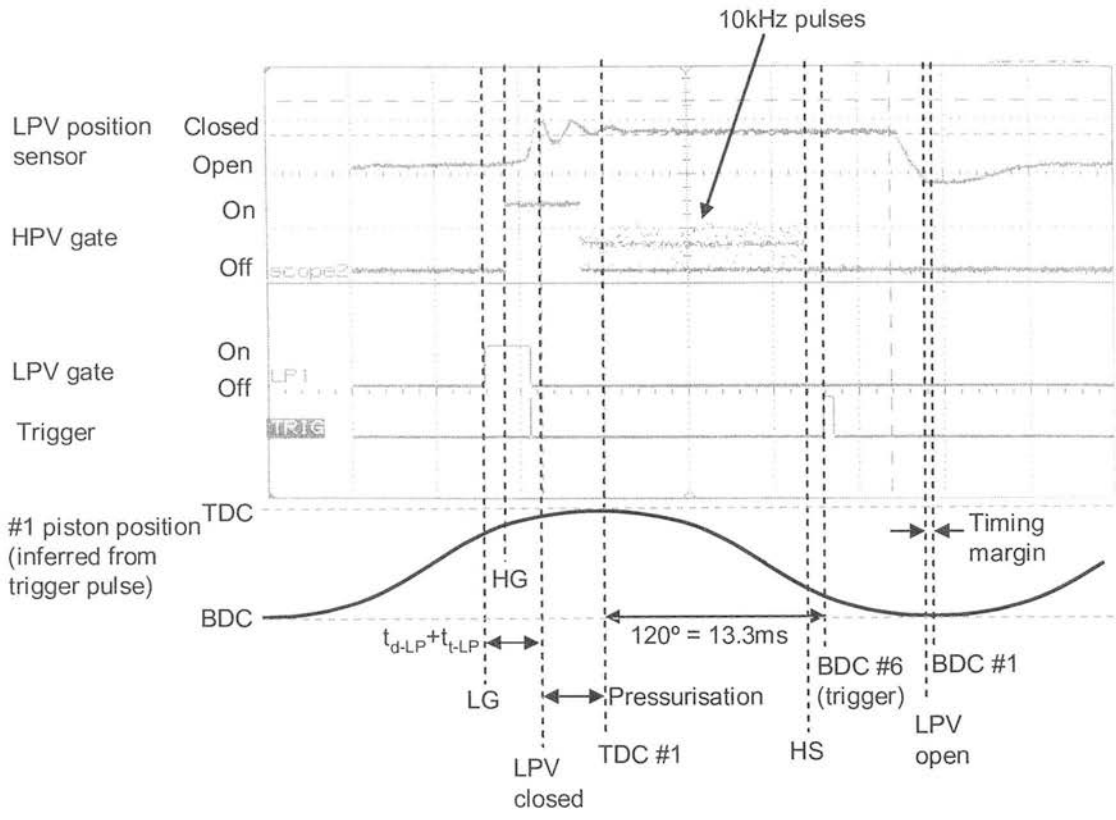


Figure 3-23: Annotated scope trace from early DDPM motoring cycle tests

The scope trace above shows the result of following the methodology given above. The "timing margin" is the time before BDC that the LPV is fully open. If the LPV is not fully open by this time, then *HP idle* cycles will take place.

While tuning the DDPM timing, it became apparent that some timing errors could cause *failure* of the DDPM cycle:

- ***LG* too late (insufficient pressurisation time).** If the LPV was closed too late, the cylinder pressure did not reach the HP port pressure before TDC. In that case, the HPV did not open because the solenoid was not strong enough to open against significant pressure. The motoring stroke did not occur and an LP intake stroke occurred instead. The machine executed *LP idle* cycles.
- **Incorrect *HG* time**. If the *HG* event occurred too early, the cylinder was not pressurised by this time and the HPV did not open. If the *HG* event occurred too late, the cylinder pressure had fallen before the HPV solenoid generated enough force to open the HPV, and the HPV did not open. The motoring stroke did not occur and a LP idle stroke occurred instead. The machine executed LP idle cycles.

- ***HS* too late.** If the HPV closed too late, the cylinder pressure did not fall to LP port pressure by the time the chamber started to contract after BDC; the LPV did not open because the LPV return force (from the spring and magnetic latch) was not strong enough to open the LPV against pressure. The intake stroke did not occur and a pumping stroke occurred instead. The machine executed *HP idle* cycles.

Some timing errors caused a *degradation* of the motoring cycle:

- ***LG* too early.** The pressurisation time exceeded that required to raise cylinder pressure to be equal to HP port pressure. Excess fluid that was pumped through the HPV before TDC reduced the net displacement of the motoring cycle.
- ***HS* too early**. The HPV closed before a full motoring stroke took place; the net displacement of the motoring cycle was reduced.

Of the failure modes, incorrect *HG* time was not a significant problem because the window of adjustment was wide and there were was no interactions with the degradations. This indicates that the actuation characteristics of the HPV were not a limiting factor to motoring cycle performance.

However, two of the failure modes and two of the degradation modes were linked:

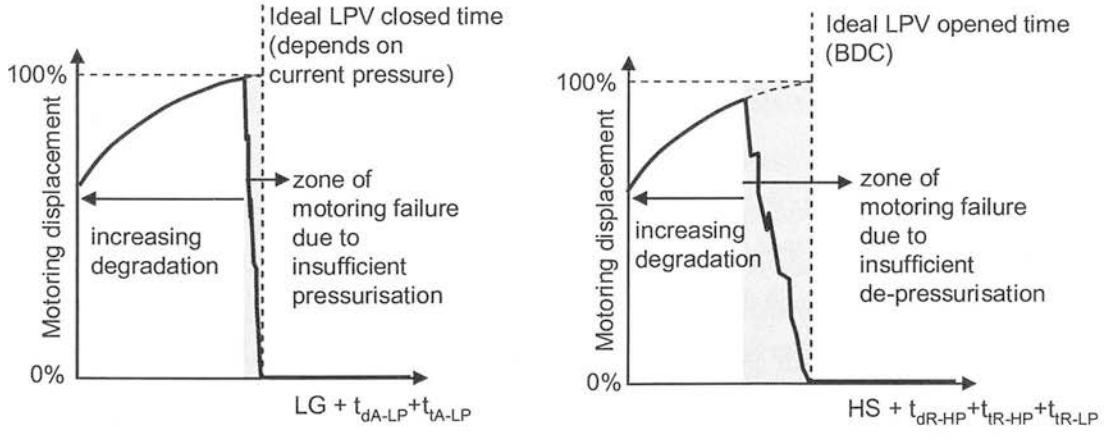


Figure 3-24: Critical zones of timing adjustment for the DDPM

The grey zone shown above represents the zone during which erratic motoring cycle failure was observed to occur. This zone was much narrower for pressurisation than it was for de-pressurisation.

The *pressurisation* sequence began with the controller firing the LPV gate – this was the LG event. After this, current rose in the LPV solenoid until the poppet started to move (t_{dA-LP} = typically 1.2ms). After the actuation transit time of the LPV (t_{tA-LP} = typically 1.5ms) the poppet hit its seat, and the cylinder pressure started to rise. The times in this sequence were

observed to be very consistent. The design of the LPV included a permanent magnet latch, which had the effect of delaying the start of motion of the poppet until a critical ampere-turns had been reached in the coil. The time to reach this critical value of ampere-turns was very stable, being simply a function of the time constant of the coil. The transit time was also fairly consistent, because by the time the poppet started to move the solenoid had already built up significant ampere-turns, and the latch force dropped off vary rapidly as soon as the armature left the latch – this gave a "snap" action to the valve. This meant that the pressurisation time could be controlled very accurately, and little reduction in effective displacement was necessary to avoid the zone of motoring failure due to insufficient pressurisation.

The *de-pressurisation* sequence began with the controller stopping the 10kHz pulses to the HPV gate – this was the *HS* event. After this, current decayed in the HPV solenoid until the return spring force exceeded the solenoid latch force and the poppet moved from the latch- the time taken for this was the HPV release delay time ($t_{dR\text{-}HP}$). This delay time was longer than most of the other delays (between 3ms and 5ms depending on the pulse duty cycle and whether the flywheel resistor was fitted) and inspection of the HP position signal showed it to be inconsistent, varying by up to 2ms depending on the HS timing. This delay was observed to shorten considerably if *HS* was earlier, and lengthen if HS was later; one explanation for this is that the HP poppet experienced significant force from the flow of fluid into the motoring cylinder. A venturi effect in the gap between poppet and seat could have caused a low pressure zone in front of the poppet. Because the back of the poppet was vented to the HP port, this low pressure zone could have caused a net closing force on the HP poppet when it was in the latched position, a force which would be a function of the flow-rate though the HPV. The HPV poppet hit its seat after the HPV transit release time ($t_{tR\text{-}HP}$). Again, position measurement in the DDPM showed this release time to be inconsistent and strongly affected by instantaneous flow-rate through the HPV, varying between 3ms and 6ms. By contrast, the transit time for the LPV to open ($t_{tR\text{-}LP}$) was observed to be consistently 1.0ms.

The result of the inconsistent valve timings for the de-pressurisation phase was that the *HS* time had to be set very early to avoid motoring cycle failure and unintentional *HP idle* cycles. This caused unacceptable degradation of the motoring cycle, which achieved only 75% of the geometric displacement of the cylinder. Although this should not be confused with "volumetric efficiency", this poor volumetric performance was seen as a very significant obstacle to the viability of the DDPM.

Initial investigations focussed on the release of the HPV. It was realised that the operation of the DDPM could be made less sensitive to the variability of $t_{dR\text{-}HP}$ by minimising this delay time as much as possible. The introduction of a resistor in series with the flywheel diode of the HPV drive circuit reduced this delay time, but at the unavoidable expense of increasing

the pulse duty cycle required to hold the HPV latched against the spring and flow forces. This is shown in the diagrams below.

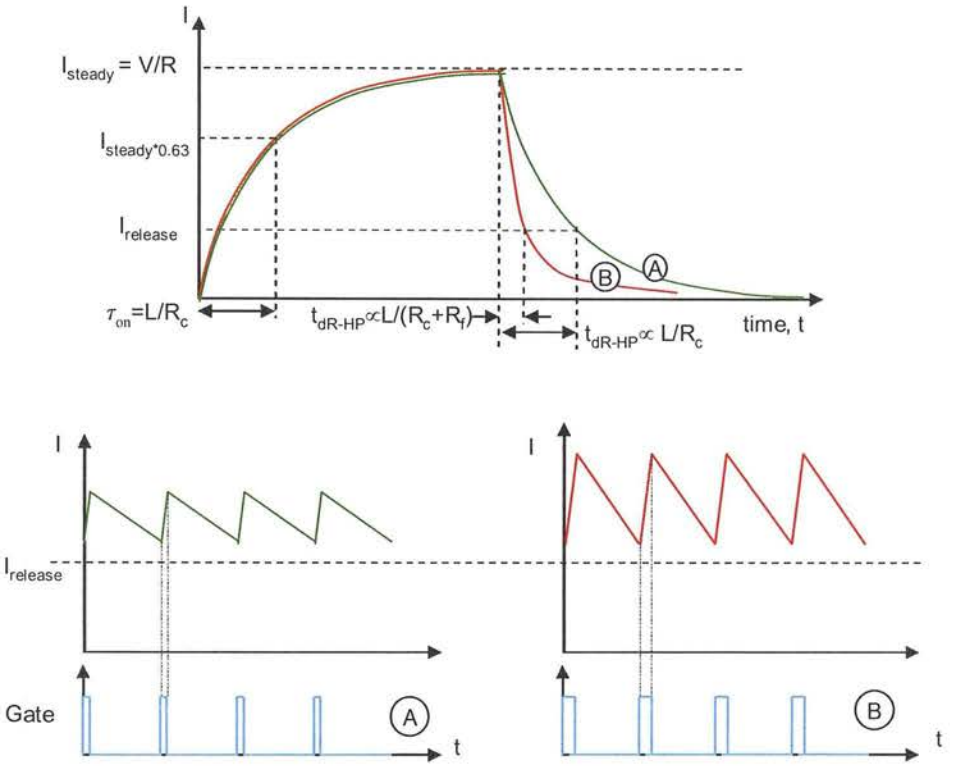


Figure 3-25: Effect on the HPV release time and pulse duty cycle with a low (A) and a high (B) value of flywheel resistance R_F

Let $i_{release}$ be the current required to hold the solenoid in the actuated position against the force of the return spring. As the value of R_F increases, the gradient of decay of the current between pulses also increases (comparing A and B above). Given that the minimum current $i_{release}$ must be exceeded by a margin at all times, this has the effect of increasing the pulse width necessary to hold the valve in the actuated position, and increasing the r.m.s. current flowing in the coil, causing increased coil heating power. This heating power imposes an upper limit on the allowable value for R_F and means that choice of R_F must be a compromise between long delay time (small R_F) and high coil power (high R_F).

A reasonable compromise for the DDPM HPVs, found by experiment, was $R_F=4 \cdot R_C$. This kept coil temperature rise below 40°C, ensuring that the insulation would survive at all operational fluid temperatures. The data below shows how this reduced the release delay considerably, but at the expense of increasing the pulse duty cycle required and therefore the power dissipated by the coil.

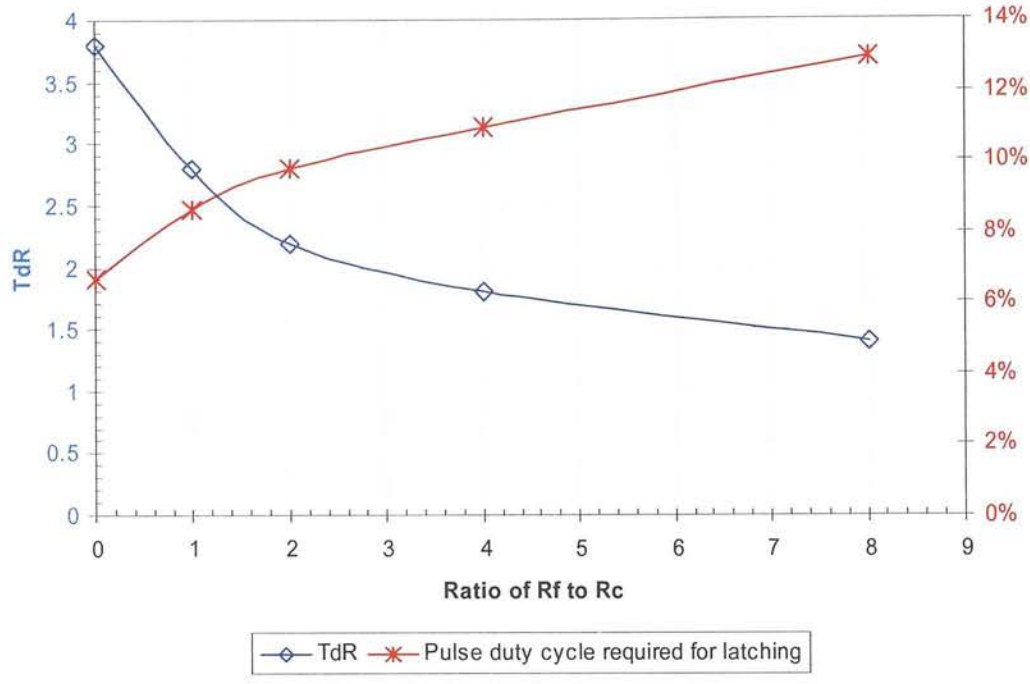


Figure 3-26: Effect of the ratio of the flyhweel resistance to the coil resistance

It should be noted that there are better electronic drive circuits which allow very fast switch-off of the latch current, without the compromise in power dissipation inherent in using a flywheel resistor (Kawase and Ohdachi, 1991). However, such circuits require two drive signals per valve from the micro-controller, which would have exceeded the digital output capacity of the prototype controller.

With $t_{dR\text{-}HP}$ minimised, it was obvious that to further increase the displacement of the DDPM, the release transit time $t_{tR\text{-}HP}$ had to be reduced. To achieve this, it was crucial to increase the strength of the return spring and minimise the moving mass. The latching current was already at the limit for coil heating, so the only way to increase the return spring force was to increase the force generated by the HPV solenoid while minimising the moving mass. This prompted the investigation with finite element analysis of the HPV solenoid magnetic circuit.

3.2.7 Conclusions of early tests of DDPM

The following conclusions were drawn from the early tests of the DDPM with measurement of valve position:

- The total HPV release delay time was too long, and fluid forces caused this time to vary in operation.

- The result was that the HS event had to be advanced a long way to avoid *HP idle* cycles and guarantee operation of the motoring cycle, resulting in reduced effective displacement of the motoring cycle.
- Inserting a flywheel resistor had a positive effect on the delay between the HS event and the start of valve release, at the expense of higher coil heating power, but had no effect on the transit time.
- To improve the release transit time, the spring force had to be increased and the moving mass minimised
- To allow increased spring force, the solenoid had to be stronger.
- Finite element analysis was required to improve the HPV solenoid.

The finite element analysis of the HPV solenoid is described in section 3.4.

3.3 Development of the LPV for the propel DDPM

3.3.1 Concept

For the reasons given in section 3.1.3, the LPV for the propel DDPM was designed to be normally closed, solenoid opened. The pressurisation event would be controlled by releasing the LPV (unlike the fixed-speed DDPM), so the release time delay of the LPV for the propel DDPM was crucial; maximum spring force at latch and minimum mass were important, as for the HPV.

For the fail-safe default to pumping mode to work effectively with an atmospheric tank, the pressure drop through the valve when closed by the spring had to be as low as possible. The requirement for maximum spring force at latch, but minimum when released, implied a high rate with little pre-load.

The magnetic circuit was subjected to parametric design improvement with finite-element magnetostatic analysis in a similar process to that described later for the HPV. Particular attention was paid to the flux concentrator because the strength of the release spring needed to be high.

In designing the geometry of the poppet, the following had to be taken into account:

- the LP intake and exhaust flow for the cylinder had to flow through the poppet;
- the armature had to be securely fastened to the poppet;
- the spring return mechanism needed a strong attachment point.

3.3.2 Poppet prototype manufacture

Injection moulding was not economically feasible for prototyping on this scale, so computer numerically controlled (CNC) machining was used. The axisymmetric outer profile was turned on a Harrison Alpha 400 CNC lathe to a DXF file generated by Solidworks 1999. The flow passages were 3D milled on a Shizuoka 4-axis machine to a G-code file generated by the NC Works CAM software, a "plug-in" for Solidworks. The small geometry necessitated a 1/16" (1.59mm) ball-ended cutter – this had to be tungsten carbide due to the abrasive nature of the carbon fibres in the PEEKCA30 material. Due to the small size of the cutter, the best spindle speed was 18,000rpm. This was achieved by mounting an electric grinding head instead of using the main spindle (which was limited to 3,000rpm). Even so, cycle time for the milling program was over 30 minutes.

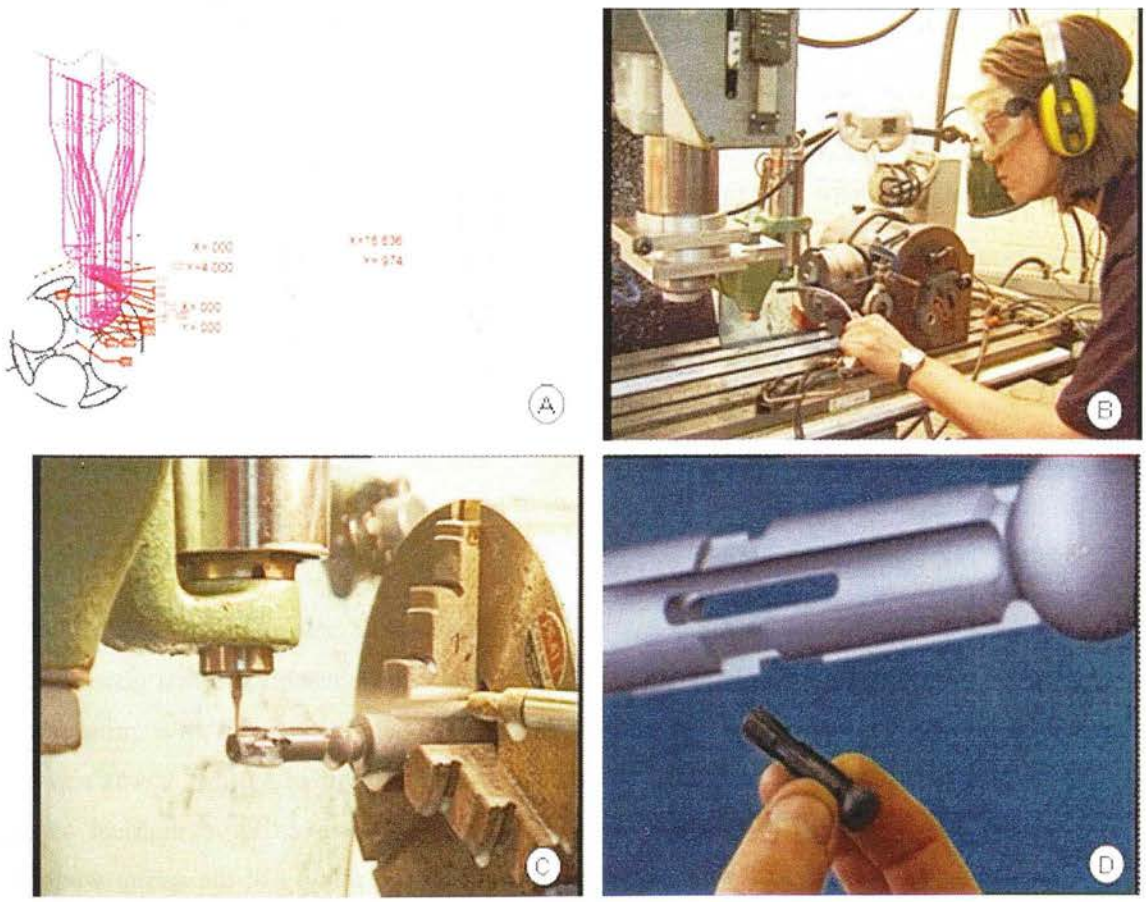


Figure 3-27: Prototyping the LPV poppet; (A) CAM planning; (B) Mill set-up; (C) Close up of machining; (D) The final part is compared to the solid model. Pictures courtesy of Jamie Taylor.

3.3.3 Mechanical design of the first-generation LPV

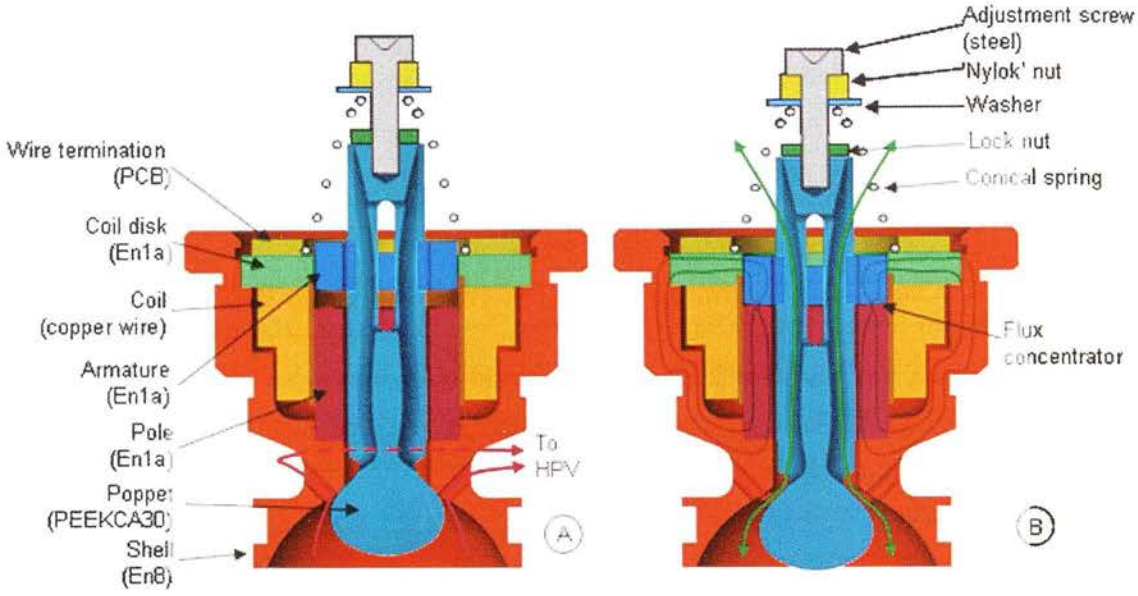


Figure 3-28: Design features of the DDPM-P LPV mk 1; (A) solenoid off, valve closed; (B) solenoid on, valve open (showing indicative flux lines)

The first attempt to satisfy the design requirements is shown above. Note that the armature is attracted downwards towards the pole face when the coil is energised. The return force was provided by a conical spring; a screw and lock-nut arrangement provided adjustable pre-load.

The armature was attached to the poppet with a "snap fit" interference design, which is detailed in Appendix 9.4.

3.3.4 The second-generation LPV

Matching the spring rate to the solenoid characteristics was crucial to achieve the fastest possible release of the LPV. Unfortunately, the conical springs used in the first generation of LPV could not be sourced in a high enough 'rate' to do this effectively. With a spring rate of 2.5N/mm, the LPV poppet exhibited a release transit time in air of 4.0ms. It was apparent that the design had to be changed to accommodate a much stiffer cylindrical spring. Calculations of the required wire diameter showed that the solidity of the spring would be such that the flow could not be ported through the spring, as intended in the first design.

In addition, early tests had uncovered a flaw with the "snap fit" arrangement. The accuracy required of the radial interference was difficult to achieve. Some combinations of armature and poppet were too tight to assemble; others were loose enough that the armature rotated in use, leading to failure of the valve.

These experiences led to a re-design of the LPV as shown below:

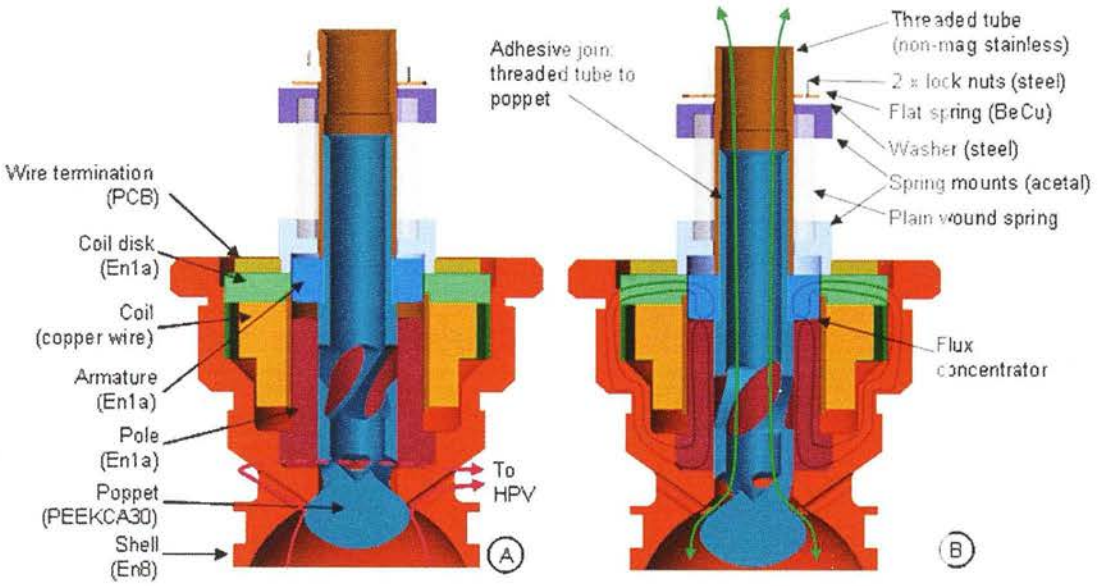


Figure 3-29: Mk.2 LPV

The poppet was now tubular, the flow going through the middle. The armature was retained by a thin stainless steel tube which was glued onto the poppet. This tube was threaded; thin locknuts allowed the preload of the plain cylindrical spring to be adjusted. Due to the poppet being tubular, there was no longer a solid core to attach the head of the poppet; instead this was attached with three "fins" which were CNC milled. The spiral cuts along the length of the poppet tube were introduced to reduce the axial stiffness of the poppet between the head and the armature- this gave the poppet the ability to absorb the impact energy when the poppet hit the seat, without causing a high stress level in the fins. Before these spirals were introduced, some fatigue failures were observed in the fins after a few hours of operation, causing 'headless poppet syndrome'.

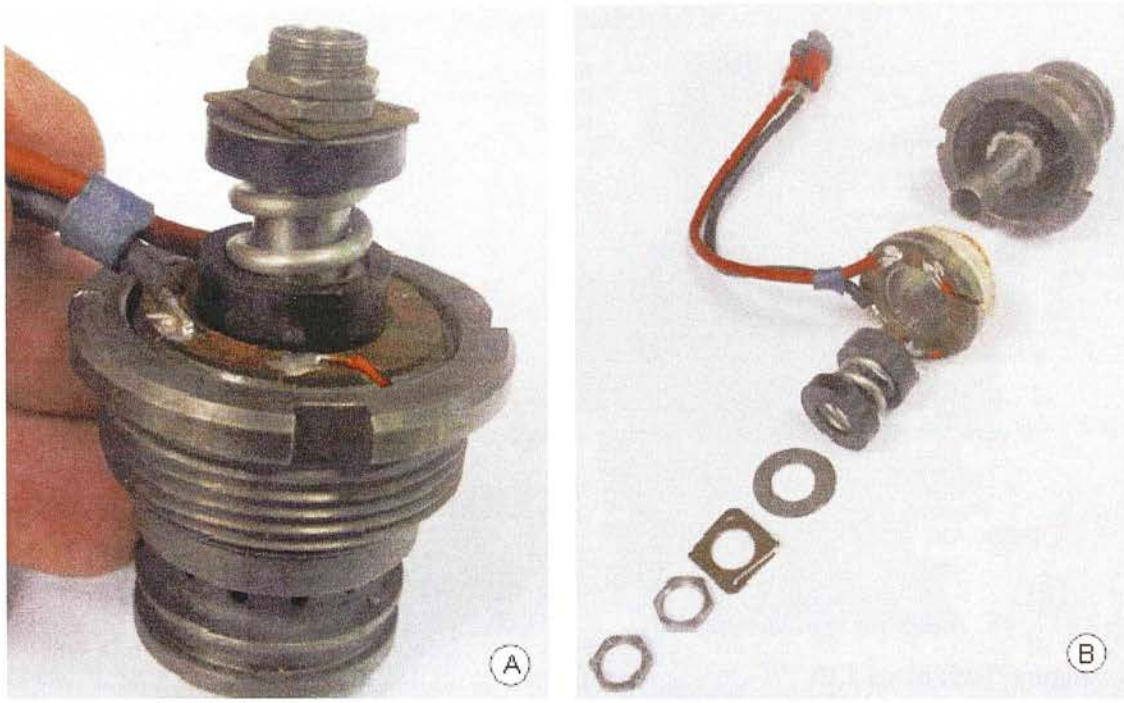


Figure 3-30: Final version of LPV mk2; (A) assembled; (B) exploded

It was important to carefully match the force from the return spring to the solenoid force available. The figure below shows the solenoid force (from FEA) and spring force, as a function of armature position.

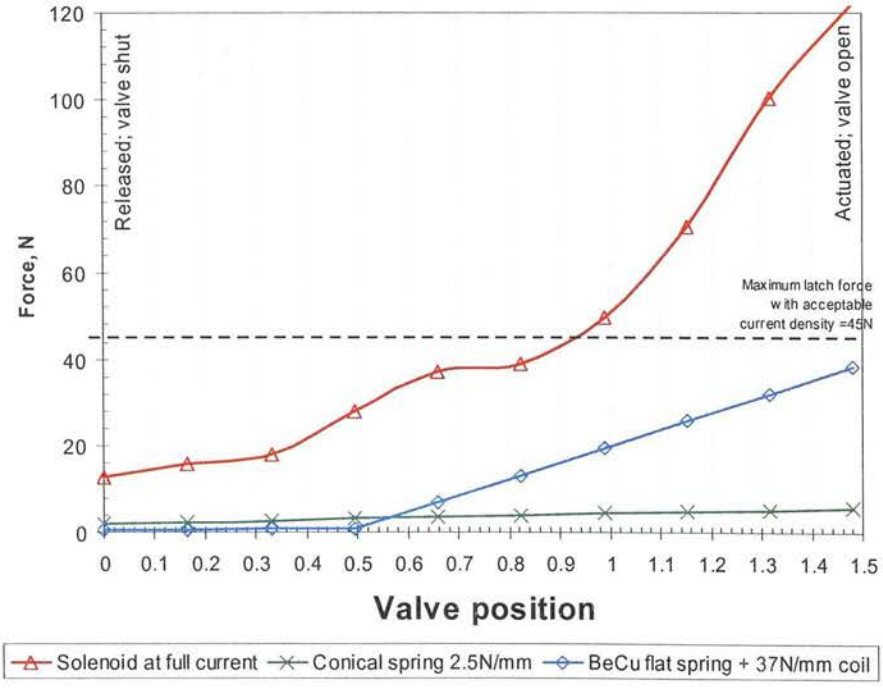


Figure 3-31: Matching the solenoid force curve (from FEA model) with the spring force

The red line in the graph above is the force from the solenoid (from the FEA model) at a current density of $22A/mm^2$ – this was lower than that achievable for the HPV because the coil had a more square aspect ratio.

The dotted line shows the maximum latch force achievable with an acceptable steady-state current density (from experiment). The green line shows the force from the original conical spring, which was obviously too weak. The blue line shows the force from the combination of a 37N/mm coil spring (with 0.5mm negative pre-load), and the BeCu flat spring. This flat spring was a washer with two thin beams, machined from 0.5mm BeCu sheet before being hardened. The flat spring provided 0.5mm of travel at 1N/mm, before becoming solid, at which point the coil spring started to be compressed. This 0.5mm of travel with very low rate allowed the propel DDPM to fill passively from an atmospheric tank in the fail-safe pumping mode. The negative pre-load of the coil spring matched the rapidly-rising solenoid force curve, providing the maximum accelerating force to the moving mass when the valve was actuated, and the maximum return force when the latch current was switched off.

3.3.5 Conclusion of the LPV development

The objective of obtaining a fast release time for the LPV for the propel DDPM was achieved. The position vs. time results (in air) from this valve are presented in Figure 3-16 p.107 (actuation) and Figure 3-17 p.107 (release). The release transit time t_{tR} of the second-generation LPV was reduced to 30% of that of the first-generation LPV, from 4.0ms to 1.2ms.

3.4 Magnetostatic finite element analysis and parametric design improvement of the HPV

3.4.1 Review of quasi-static solenoid analysis methods

The design of solenoids is traditionally done by considering the operation at a static or quasi-static operating point (Roters 1945). Such methods ignore the transient effects which occur during movement of the armature, but allow the basic geometry to be designed for a certain limiting condition, such as to achieve a certain force at a certain current and travel. However, even once transient effects are removed, the static analysis of solenoids is not straightforward.

For quasi-static conditions, the simple relationship between current, position and force in a solenoid presented in eq. 41 (page 99) is only valid under the following conditions:

1. All the magnetic flux flows through the axial air gap, the area of which remains constant.
2. The permeability of the materials involved is constant, so that B is proportional to i.

Even if the second condition holds, the first condition is not valid in practical long-stroke solenoid designs.

Magnetic flux follows whichever path offers the least resistance to its flow. Even in the linear region of the magnetic materials, this means that when the axial gap is large, the flux tends to follow alternative paths other than the main axial gap. This flux does no useful force-generating work and so is termed "leakage" (Roters 1945, p.215). Leakage is strongly affected by the reluctance of the circuit and hence the position of the armature.

As the axial gap increases, the aspect ratio of the gap changes, causing "fringing" of the flux between the axial pole and the armature (Roters 1945, p.140). This reduces the flux density compared to what would be expected if the flux only flowed normal to the pole, thus reducing the force produced.

The second condition above is not valid for practical high-force solenoid designs. All magnetically soft materials (i.e. materials which are usefully conductive of magnetic flux without remanence) are non-linear, in that the permeability changes with respect to the flux density. At medium flux density in mild steel, i.e. below 1.0 tesla (T), this effect can be neglected and the linear equations are useful (Roters 1945 p.221). However design pressure to squeeze the maximum force from the smallest actuator pushes the designer to maximise flux density. Between 1.0T and 2.0T, the permeability of mild steel decreases dramatically as shown below.

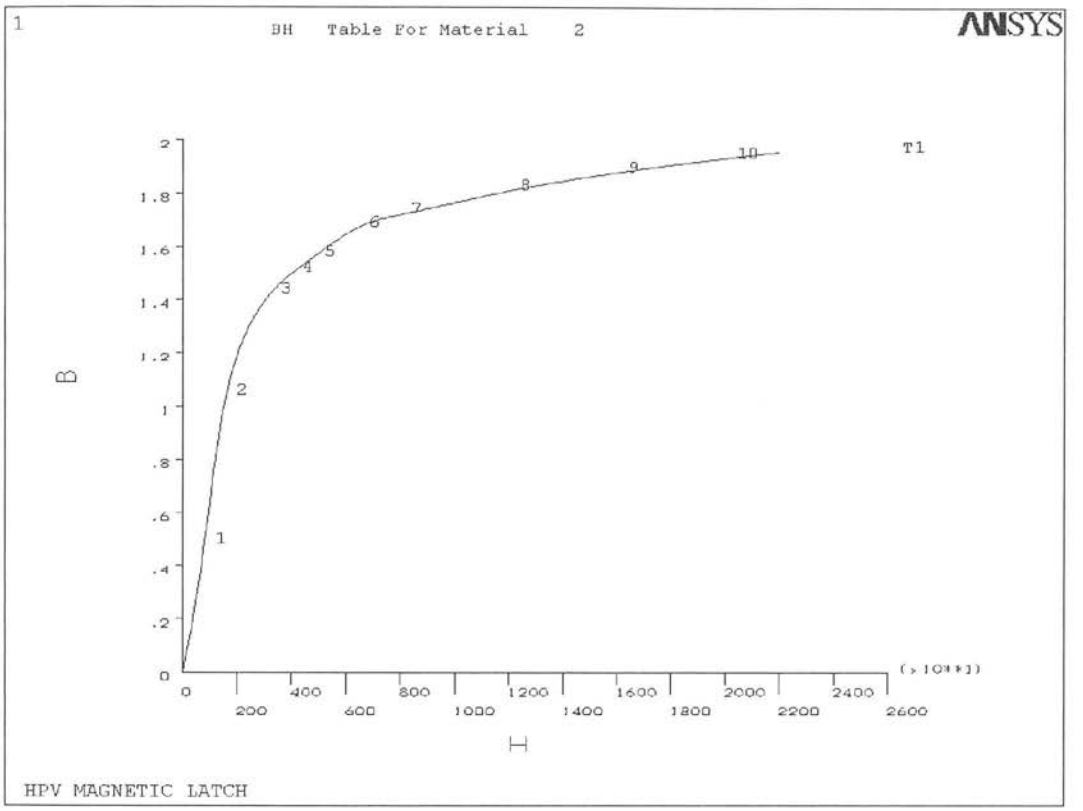


Figure 3-32: BH curve for EN1a mild steel

The curve above shows the relationship between ampere-turns per meter (H) and flux density (B) for EN1a mild steel, as used for the later FEA analysis. The gradient of this line is the "incremental permeability" of the material. Between the origin and point 1, the material can be considered to be magnetically linear with an incremental permeability some 476 times higher than air. However between point 9 and 10 the incremental permeability has fallen to 10.9 times that of air. Above this point it can be modelled as having an incremental permeability equal to air (Woolman and Mottram 1966) – once the flux has reached this level, the material is "saturated".

Looking at eq.41, it is natural to design the magnetic circuit to concentrate the flux in the axial gap to the maximum extent, such that the force is also maximised, because force is proportional to the square of the flux density. This concentration of flux tends to push the magnetic materials into the saturation region, causing the flux distribution to change due to leakage.

Together the effects of leakage, fringing and saturation cause the magnitude and distribution of magnetic flux, and therefore the force on the armature, to vary substantially as the armature moves and as the current in the coil changes. Solenoids which are used for proportional duty may be specially designed to be magnetically linear, at least around an operating point.

However, this is at the expense of the absolute magnitude of force (Roters 1945). Tightly-packaged on/off solenoids where linearity is not important are highly non-linear in operation. Direct analytical solutions do not exist, so numerical techniques are necessary.

Roters (1945) gives a comprehensive introduction to numerical approaches to deal with the non-linearity of magnetic materials, leakage and fringing. These involve breaking the problem down into a series of geometrical primitives, each of which are analysed separately to find the reluctance of the entire magnetic circuit. The current is applied in iterative steps, and at each step the incremental permeance of each primitive element is calculated with reference of to the B-H curve of the material, the magnetic circuit being treated as a resistance network. Roters demonstrates good accuracy for this technique, as long as the geometry of the air gaps does not change significantly as a function of stroke; although without the digital computer, this approach was extremely time-consuming.

Roters notes that there are no known direct analytical solutions of most practical solenoid design problems and this continues to be the case. Using formidably complex mathematics borrowed from gravitational physics, Conway (2001) described a general analytical method for mapping the field in axisymmetric solenoids. As the method cannot deal with ferromagnetic regions, it is of limited use for solenoid actuator design.

Since Roters' work, the digital computer has revolutionised the analysis of solenoids. The finite element method using the magnetic vector potential formulation has become the standard for analysing solenoids. However, even with the power of modern computers, 2D approximations (whether axi- or planar- symmetric) continue to be preferred over 3D solutions because of the much reduced computer resources and time required (Prieto et al. 1999).

3.4.2 Rampen's work on solenoid analysis

By the time Rampen (1992) was writing, the ANSYS software package for PCs could be applied to magnetics problems. However Rampen notes that, in the way he used it, ANSYS was not readily suited to iterative design improvement of the magnetic circuit of the LPV. He imported the geometry into ANSYS via a DXF file, generated by a 2D CAD program. While this minimised the amount of initial work to get a single solution, the method made design improvement extremely difficult because the geometry description was not parametric. Design changes had to be implemented in the CAD package and the DXF file re-exported at each design iteration.

Such was the difficulty and time taken to execute a single FE solution, Rampen's approach to design improvement was to simplify the circuit into a set of algebraic simultaneous equations

which he solved numerically. He created an equivalent magnetic circuit and solved the magnetic equations for each element of the circuit (armature, pole, coil and air gaps) in the way described by Roters (1945) but with a computer instead of hand calculations. Magnetic saturation was dealt with by the use of a look-up table. By iterating this model through a sweep of design variables, Rampen was able to get to what seemed to be an optimum design.

However this approach was based on a number of key simplifications:

1. There was no detailed model of flux leakage. Rampen acknowledges that the model uses a constant to describe this behaviour which can only be guessed at within a factor of two. In fact this leakage behaviour is highly sensitive to armature position and level of saturation of the magnetic circuit

2. Rampen's equations of magnetic saturation took no account of the detailed geometry of individual components, which causes variation of the effective cross-section of steel normal to the direction of flux. For instance, in axially thin parts in which flux is radial, the flux path is divergent and hence the flux density decreases as it flows further from the axis. This means that the material may saturate at a small radius but not at a large one.

3. There was an assumption that the flux density in an air gap is uniform. In fact the flux density in the axial air gap can be expected to be non-uniform due to fringing. In the latch condition this effect is minimal because the air gap is small compared to the other dimensions. However in the case of the actuating air gap this is not true. Hence Rampen's model can be expected to give an over-estimate for the actuating force produced.

Rampen's approach has the advantage of requiring little computer power to explore the design parameters in a reasonable amount of time, which at the time he was writing was a major consideration. By the time the author was working on the subject, both ANSYS and computer hardware had progressed enough that a parametric FE model running on a PC was considered feasible, which would not suffer from the drawbacks above.

3.4.3 Overview of the parametric FEA model

It was intended from the outset that the FEA model of the HPV would be defined parametrically. This meant that the geometry would not be imported from a CAD package as Rampen did, but instead would be constructed automatically from a set of equations, for each design iteration. This would allow the investigation of the effect of individual design parameters or combinations of them.

It was obvious that many iterations of the geometry would have to made before the optimum was found. It was therefore important to minimise the amount of calculation necessary to assess each design iteration.

The valve solenoids studied are overwhelming axisymmetric and therefore this was the first simplifying assumption made, to reduce the complexity of the geometry and maximise the speed of each solution.

The operation of the HPV solenoid is inherently transient, because the current in the coil varies significantly during the motion of the armature. A fully detailed model would have to allow the armature to move during the simulation, causing motional e.m.f. which would affect the current in the coil. The motion of the armature would result from solving the one-dimensional equation of motion of the poppet, taking into account magnetic force, mass, spring and damping. Such simulations are not possible within ANSYS and involve a large amount of computation (perhaps measured in hours) for each design iteration to model the time series of current and armature motion. This is in conflict with the goal of rapid design improvement of the magnetic circuit.

So to further simplify the problem, each design iteration was instead assessed in the two most important quasi-static operating points, at which the armature is stationary; these were the "grab" and "latch" cases, which are defined in detail below.

Together, these simplifying assumptions allowed the calculations needed to assess each design iteration to be limited to around 30 seconds on a 1GHz PC.

3.4.4 Design constraints for design improvement

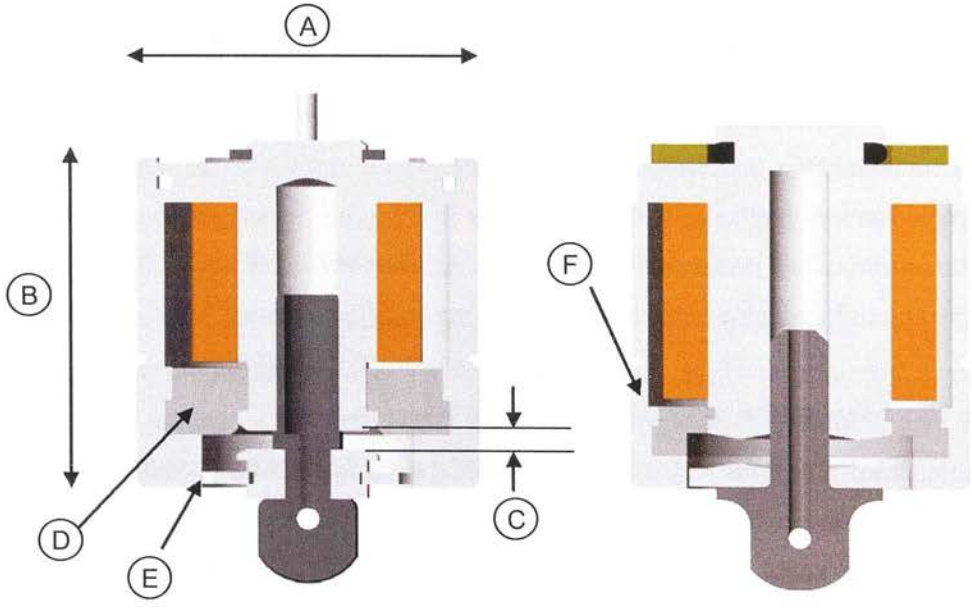


Figure 3-33: Design constraints for the HPV design improvement; left=baseline, right=improved

The figure above shows the basic design constraints for design improvement. The fundamental constraint was that the HPV had to fit within the existing cavity in the pump ring. This fixed the outside diameter (A) and the overall length (B) of the valve cartridge.

The travel of the valve poppet (C) for the baseline design was 2.0mm, which gave an adequate pressure vs. flow characteristic when the valve was open. There was no need for improvement here, and any increase in travel would adversely affect the grab force of the solenoid, so the 2.0mm travel was fixed for the design improvement.

The pressure bulkhead (D) and its interfaces had to withstand a hydrostatic force. Hand calculations were done as the design improvements progressed to ensure that the shear stress in the bulkhead and the inside and outside retaining steps was acceptable.

The radial gap (E) of 0.15mm of the baseline design was thought to be the minimum allowable. If this was too small, angular misalignment caused by clearance in the poppet guide could cause the flux pattern in this gap to be non-axisymmetric, causing a side force on the poppet which could have resulted in excess friction. This value was fixed during the design improvement process.

The wall thickness (F) of the cartridge shell was limited by the stress in the M24x1.5mm thread and the retaining step of the bulkhead. Stress calculations showed there was capacity in the baseline design for the wall thickness to be reduced.

3.4.5 Parameterisation of the solenoid geometry

A section of the improved HPV geometry is shown below.

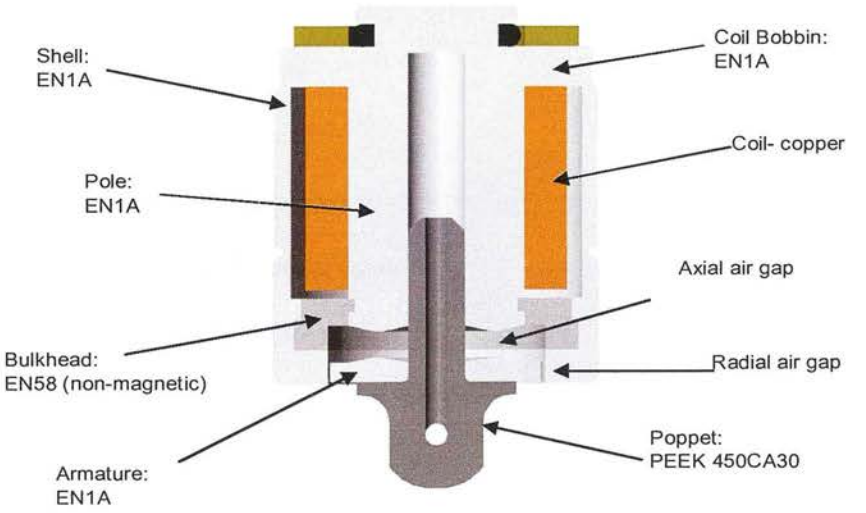


Figure 3-1: Section of the improved HPV showing materials and nomenclature

The following procedure was followed to create a parametric geometry description.

1. A paper sketch was made of the section of the magnetic circuit, comprising points joined by lines. Copies of this sketch were annotated with a number assigned to each point, line and area as required by ANSYS. It was crucial to retain control of the numbering of the geometric entities for subsequent application of mesh density, boundary conditions etc. This was greatly complicated by the limitation that lines and areas in ANSYS are automatically assigned numbers when they are created, and these numbers cannot be specified; therefore the order in which the entities were declared in ANSYS had to match the numbers assigned in the sketch.
2. The parameters required to be varied were noted on the sketch as dimensions with variable names instead of numbers.
3. The coordinates of each point were described in terms of the parameters by equations, and these equations were used in the definition of each key point.

The figure below shows the definitions of the important parameters, and an example of a matching geometry definition script.

```
*set,axgap,0.00001      ! set gap between lower and moving pole
*set,rgap,0.00015       ! radial clearance
*set,thk,0.0015         ! set moving pole web thickness
*set,coilir,0.0065      ! Coil internal radius
*set,poleir,0.002       ! Moving pole internal radius
*set,poleer,0.0065      ! Moving pole external radius
*set,ringir,0.0095      ! Inside radius of radial flux ring
*set,taperw,0.0003      ! Width of tapered part of moving pole face
*set,taperh,0.00035555  ! Height of tapered part of moving pole face
*set,thinmag,0.003      ! Thickness of thin magnetic poleface
```

```
K,1, 0.0, 0.0, 0.0
K,2, 0.008, 0, 0
K,3, 0.008, 0.002, 0
K,4, 0.010 , 0.002 , 0
K,5, ringir , 0.0075-axgap-thk , 0
K,6, 0.010 , 0.0075-axgap-thk , 0
K,7, 0.014 , 0.0075-axgap-thk , 0
K,8, ringir , 0.008-axgap , 0
K,9, 0.01 , 0.008-axgap , 0
K,10, 0.011 , 0.015 , 0
K,11, 0.011 , 0.026+magl , 0
K,12, 0.014 , 0.029+magl , 0
K,13, 0.00 , 0.029+magl , 0
K,14, 0.00 , 0.026+magl , 0
K,15, magr , 0.026+magl , 0
K,16, coilir, 0.026+magl, 0.0
K,17, coilir, 0.026, 0.0
K,18, magr, 0.026, 0.0
K,19, 0.0, 0.026, 0.0
K,20, 0.0, 0.025, 0.0
K,21, poleir, 0.0233, 0.0
K,22, poleir, 0.0103+THINMAG, 0.0
K,23, poleer, 0.0103+THINMAG, 0.0
K,24, poleir, 0.0103-axgap-(taperh/2), 0.0
K,25, 0., 0., 0.0
K,26, poleer, 0.0103-axgap-(taperh/2), 0.0
K,27, poleer, 0.0075-axgap, 0.0
K,28, ringir-rgap, 0.0075-axgap, 0.0
K,29, ringir, 0.0075-axgap, 0.0
K,30, ringir, 0.0075-axgap-thk/2, 0.0
K,31, ringir-rgap, 0.0075-axgap-thk/2, 0.0
K,32, ringir-rgap-thk/2, 0.0075-axgap-thk, 0.0
K,33, poleer, 0.0075-axgap-thk, 0.0
K,34, poleer, 0.0055-axgap, 0.0
K,35, poleir, 0.0055-axgap, 0.0
K,36, 0.0, -0.003, 0.0
K,37, 0.014, -0.003, 0.0
K,38, coilir, 0.015, 0.0
K,39, poleir, 0.0103+(taperh/2), 0.0
K,40, poleer, 0.0103+(taperh/2), 0.0
K,41, poleir+((poleer-poleir-taperw)/2), 0.0103-axgap, 0.0
K,42, poleer-((poleer-poleir-taperw)/2), 0.0103-axgap, 0.0
k,43, poleir+((poleer-poleir-taperw)/2), 0.0103, 0.0
k,44, poleer-((poleer-poleir-taperw)/2), 0.0103, 0.0
```

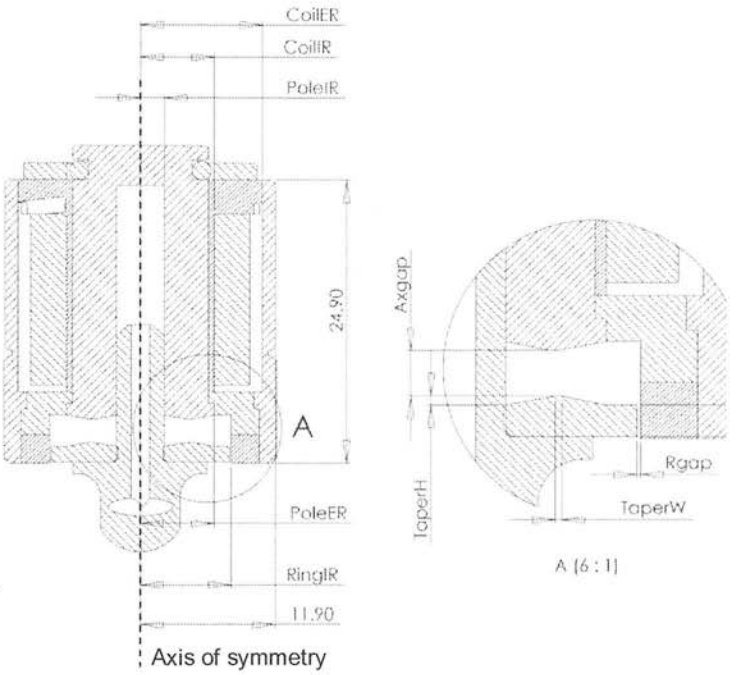


Figure 3-34: Parameterised geometry of HPV (final improved geometry shown)

The important geometric parameters, labelled in the figure above, are described below:

coilir: The internal radius of the coil

coiler: The external radius of the coil.

poleir: The internal radius of the poppet guide bore in the central pole.

poler: The external radius of the central pole.

ringir: The internal radius of the radial flux ring.

rgap: The radial gap between armature and the radial flux ring.

taperw: The width of the flux concentrator.

taperh: The height of the flux concentrator.

The position of the armature was controlled by *axgap*; this defined the axial gap between the pole and the armature. The y-coordinates of all the points making up the armature were therefore a function of *axgap*.

3.4.6 Implementation in ANSYS 5.4

ANSYS 5.4 (ANSYS Inc. 1997) is a finite element analysis package which runs on a standard PC. It is a very mature product which has its roots in FORTRAN code written for mainframes in the 1970's. Although a graphical user interface has been added, much of this old code remains behind the scenes. Initially it seemed easier to define geometry using this graphical interface, but this was quickly found to be cumbersome. A more controllable method was to create a script of commands with a text editor and feed these into the ANSYS command line. When used in this way ANSYS offers powerful abilities to parameterise any numerical input, allowing investigations to be carried out with varying geometry (for instance, the diameter of the armature), boundary conditions (such as the current density in the coil) or analysis controls (such as the mesh density on a particular line).

The input file to create the geometry and execute a single solution is given in Appendix 9.6. It consists of the following parts:

Geometry. Points were placed in the 2D plane according to equations of the geometric parameters. These were joined by lines, and the lines joined to create areas.

Material definitions. The non-linear saturation of EN1a mild steel was implemented in a table of H and corresponding B values. Above B=1.935 the incrementally permeability was assumed to be equal to air (Woolman and Mottram 1966). ANSYS interpolates this table during solver iteration. Although a separate material definition was created for the EN58 non-magnetic stainless steel used in the bulkhead, this was assumed to have the same permeability as air (Woolman and Mottram 1966), so was functionally identical.

Boundary conditions. Each of the areas was assigned a material. A current density excitation around the axis was applied to the area of the coil. The lines at the boundary of the model were given a "flux parallel" boundary condition; it was assumed ensured that flux did not leave the meshed area. The "FMAGBC" macro was executed on the areas representing the armature to allow calculation of the force after solution.

Mesh. The areas were meshed with "Plane13" axi-symmetric quadrilateral elements. Element size was specified along the lines bounding areas of highest magnetic potential gradient such as the axial gap, to give higher mesh density, and therefore higher accuracy, in these regions. The remaining areas were meshed using the ANSYS automatic mesh generator. At each major design change, a global mesh refinement was executed, and the change in magnitude of

the force on the armature was inspected; the mesh was refined until force changed by less than 1%; at this point the mesh was assumed to be sufficiently fine.

Solve. The magnetic vector potential method was used (Ansys Analysis Guide 1997), the default method for 2D static analysis. The default iteration tolerance of 0.001 was used, and the solver typically converged after 5 iterations, taking about 30 seconds on a 1GHz PC.

Post processing. Pre-defined macros were used to extract results after the solution was calculated. The macro FMAGSUM calculated the force on the armature, and SRCS was used to calculate the inductance of the coil.

The figure below shows the ANSYS geometric entities and resulting mesh created by the script file.

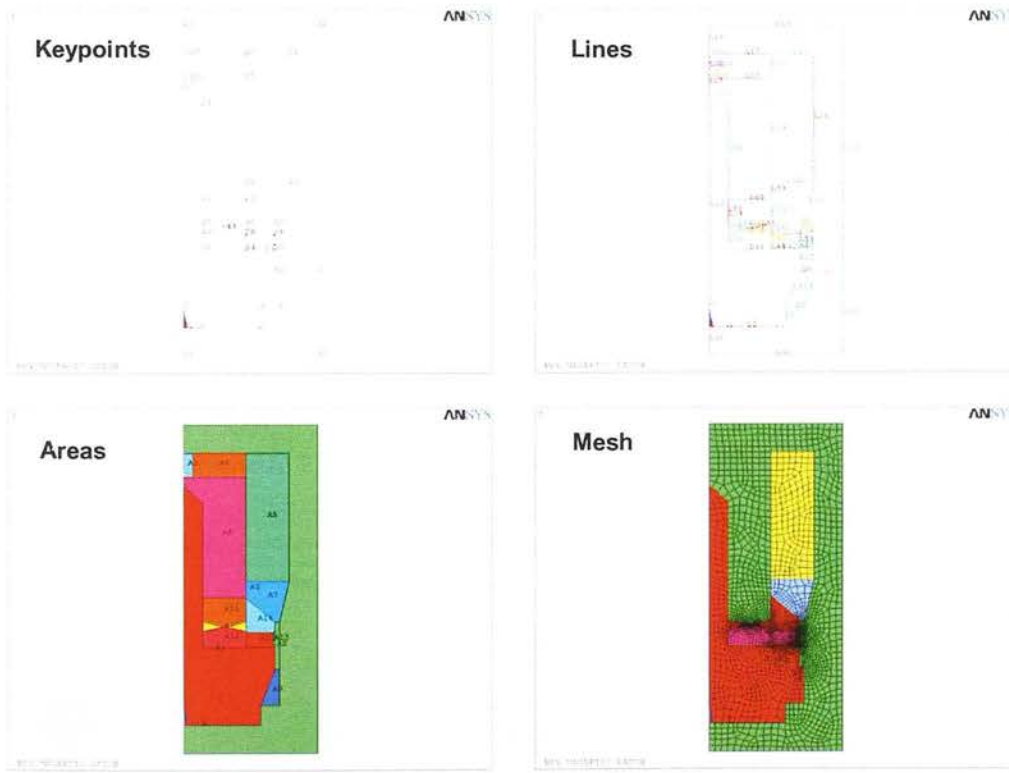


Figure 3-35: Axi-symmetric geometry definition in ANSYS (Y is the axis of symmetry). Mesh coloured by material: red=air, green=steel, yellow=copper (coil), blue=non-magnetic stainless steel, pink=armature (also steel)

The definition of the parameters of the FEA model is given below.

Name	Description	Type	Value before optimising	Value after optimising
ringir	Radius of the radial gap	Geometry	7.5mm	8.0mm
thk	Thickness of the armature	Geometry	0.9mm	1.4mm
coilir	Internal radius of the coil	Geometry	4.0mm	6.5mm
coiler	External radius of the coil	Geometry	10.0mm	10.75mm
poleir	Internal radius of the pole	Geometry	2.5mm	2.1mm
poler	External radius of the pole	Geometry	4.0mm	6.5mm
taperw	Width of the flux concentrating taper	Geometry	-	0.5mm
taperh	Height of the flux concentrating taper	Geometry	-	1.0mm
axgap	Axial gap between pole and armature – defines armature position	State	0-2mm	0-2mm
current	Current density in the coil	State	-	-

3.4.7 The design improvement process

ANSYS 5.4 has a module called Design Optimisation (ANSYS Analysis Guide 1997). In theory, this allows the automatic optimisation of a design to minimise a given "objective function".

Preparation of an automated design optimisation proceeds in the following phases:

1. A parametric FEA model is created. Great care must be taken with the choice of parameters to ensure that the topology of the model does not change with any combination of design parameters e.g. by lines crossing.

2. An objective function is defined, which is to be minimised. In the case of the HPV solenoid, this could be some weighted combination of the reciprocal of latch and grab forces.

3. A number of the parameters of the model are designated as "design variables" (DVs); each represents a degree of freedom available to the optimiser. Each DV is allowed to vary within user-definable limits.

4. Results of the analysis can be defined as "state variables" (SVs). Each SV has an allowable range.

When the design optimisation is started, the optimiser seeks to minimise the objective function subject to the condition that all of the SVs are within allowable limits, using a so-called "hill-climbing" search algorithm.

Automatic optimisation been reported in the academic literature for the problem of solenoid design. Mathew and Hippner (2004) investigated a highly simplified plunger solenoid, described by only three parameters, which they had no intention of making or using. Genetic algorithms were used to "breed" design candidates, each of which was subject to FEA and assessed against the objective function – in this case, maximising force per unit volume of the solenoid. The result was seen to converge on an optimum after 120 generations, but there is no consideration by the authors as to *why* this was the optimum. They conclude that "a fully automated optimisation system…has been successfully implemented".

The principle of automatic optimisation is described graphically in the figure below.

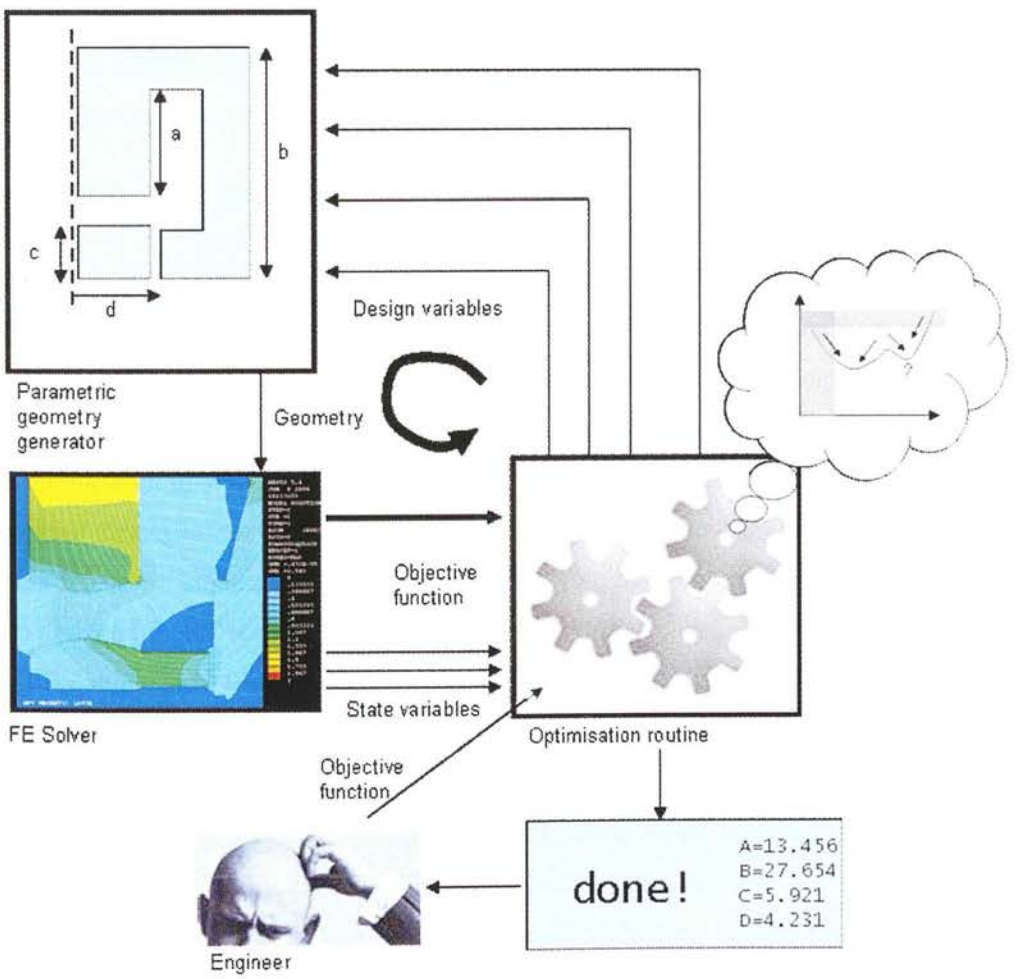


Figure 3-36: "Automatic" design optimisation method

Initially it was thought that automatic optimisation would be very useful, but attempts to get intelligible results were frustrated by the following factors:

- It was difficult to encode the design objective of a "better solenoid" into a mathematical function. At the start of the optimisation process the trade-offs that would have to be made later were not clear.
- It was difficult to encode the non-magnetic constraints of the design (e.g. manufacturing considerations, fluid flow, mechanical stress) as state variables (SVs), as these were outside the domain of the solver. It was realised that a truly "fully automatic" design optimiser would have to include an expert system representation of all the engineer's knowledge in all domains. It was realised that ANSYS does physics – it does not do engineering.
- Some parameters are quantised in practice (Pettersen et al. 2005) e.g. available tool diameters. All design variables (DVs) are assumed by ANSYS to be continuously-variable.
- Most important, once the "optimum" was found, the engineer was left with no insight as to *why* it was the optimum. When asked by a superior why a particular dimension has to be changed to 23.34mm, the answer "the computer says so" is not sufficient.

As a result of this early experience, automatic optimisation was rejected, and an interactive, iterative approach was followed. From the initial baseline result, all but one of the parameters were fixed and the remaining parameter swept through a permissible range. For each solution, plots of flux distribution were created by the script file and these were joined together at the end of the sweep into a video file, revealing how the flux distribution changed in response to the chosen parameter. Plots of force as a function of the swept parameter were also created. Together these give insight into the behaviour of the magnetic circuit and an intuitive understanding of the compromises involved in the design. This was repeated with each of the parameters in turn, sometimes returning to parameters already studied to see if the old conclusions were still valid with the new parameter values. At each stage the results were assessed and checked against the non-magnetic constraints such as ease of manufacturing.

This interactive approach is shown graphically below.

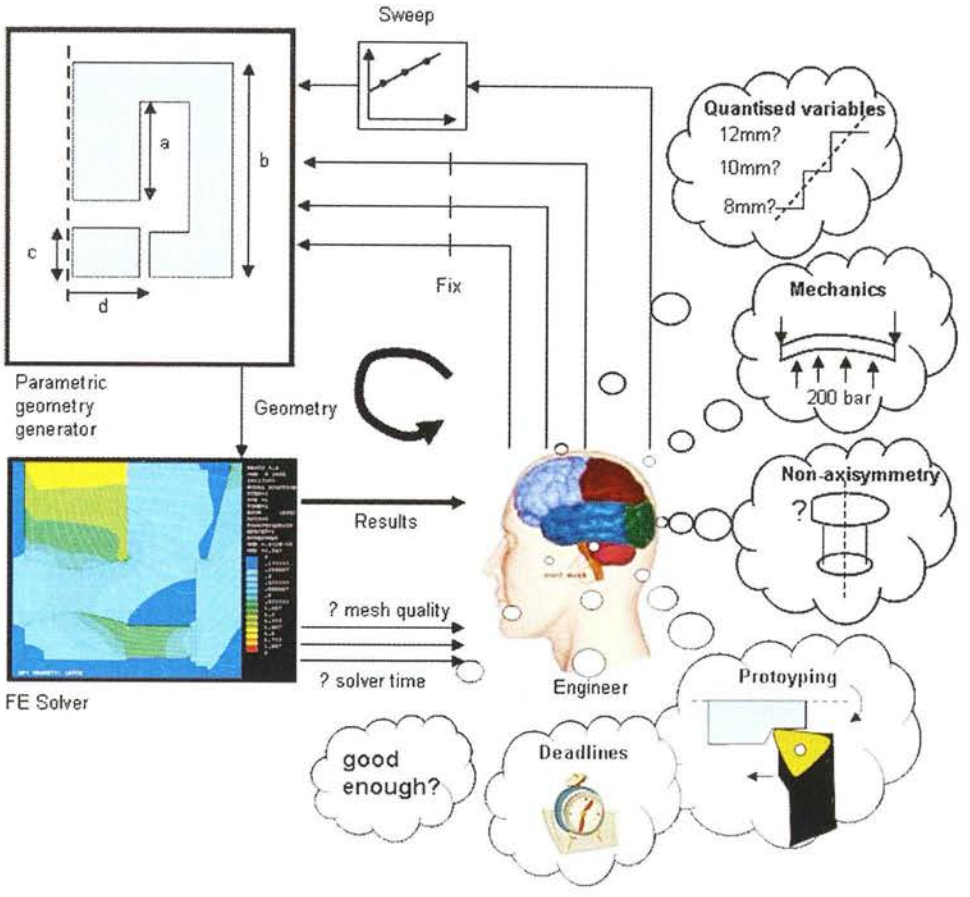


Figure 3-37: An engineering approach to "optimisation"

The result of this process was that the engineer gained an insight into the trade-offs that were being made at each stage of the 'optimisation' process. Each of the design changes could be justified by a narrative explaining the cause-and-effect relationships.

3.4.8 Design improvement of the HPV solenoid

Once the parametric FE model was created, the solenoid design could be investigated. Both coil current and armature position were parameters, so it was possible to investigate the operation of each candidate geometry in any possible steady-state. In order to simplify the comparison of competing designs, two load cases were defined which captured the essence of the desired functionality:

"Grab" force: Axgap = 2mm, current density = 25 A/mm^2. In order to fit the strongest possible return spring, it was important to know for each design the maximum force capable of being generated to actuate the armature, when the air gap was at a maximum (axgap=2mm). A maximum current density needed to be established which kept the temperature rise of the windings (due to resistive heating of the coil) within acceptable limits. The HPV had a heat conduction path through the oil which surrounded it, and experiments on

the bench revealed that the HPV coil could run at a steady 8A/mm^2 with 40°C temperature rise. However the HPV has to work at only a 10% duty cycle (4ms out of 40ms for a DDPM at 1500rpm) so it was acceptable to "overdrive" the coil by a factor of 10 on power, or a factor of 3.16 times for current, giving an acceptable current density for the grab case of 25A/mm^2.

"Latch" force: Axgap=0.01mm, current density = 1.25A/mm^2. During the motoring stroke the HPV needed to be kept in the open (actuated) position, with the armature in contact with the pole. This presented a problem because the armature must always be surrounded by air elements in order for the solenoid force to be calculated, and the smaller the axial gap gets in the FEA model, the more elements are needed to resolve the flux pattern in the gap. A gap of 0.01mm was chosen, which gave a force only 1% lower than a gap of 0.005mm but with substantially fewer elements needed[2]. A current density of 1.25A/mm^2 was chosen for this load case because it must be possible to maintain this current indefinitely without overheating (in the case of a propel motor operating close to zero speed) and it was desired to minimise the electrical power consumption.

With the two load cases defined, preliminary investigations were made on the baseline design. The following figures show plots of the flux density and the flux lines for both the "grab" and "latch" cases.

[2] When the motor operates at 1500rpm, the valve is only latched for 20ms. It is likely that the oil squeeze film effect prevents the gap closing completely during this time, so 10 microns may be more realistic than 5 microns.

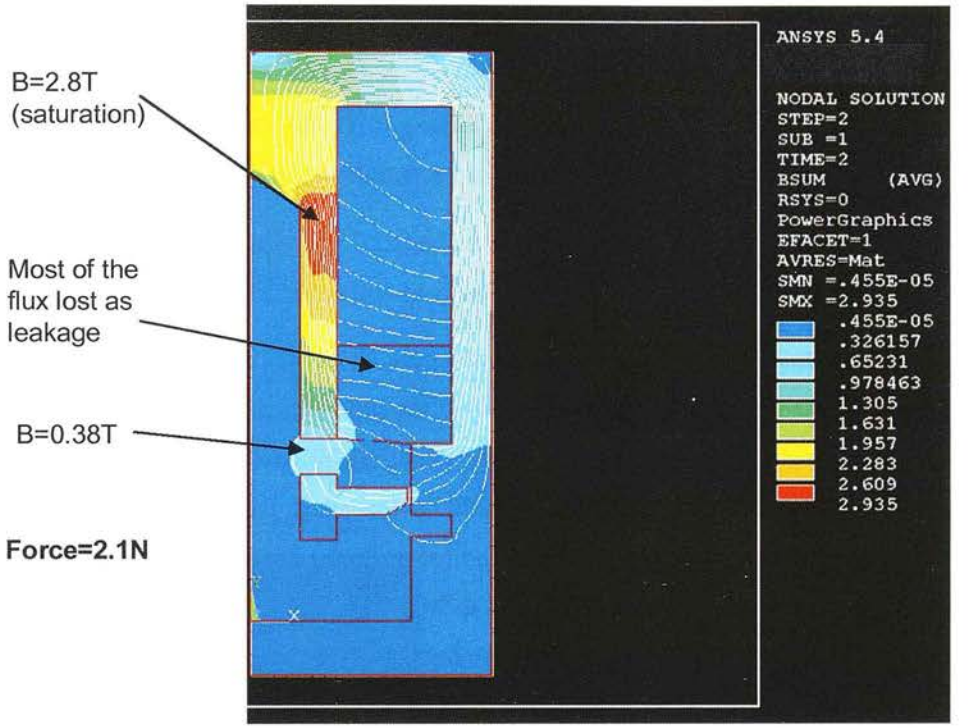


Figure 3-38: Baseline design; "Grab" case

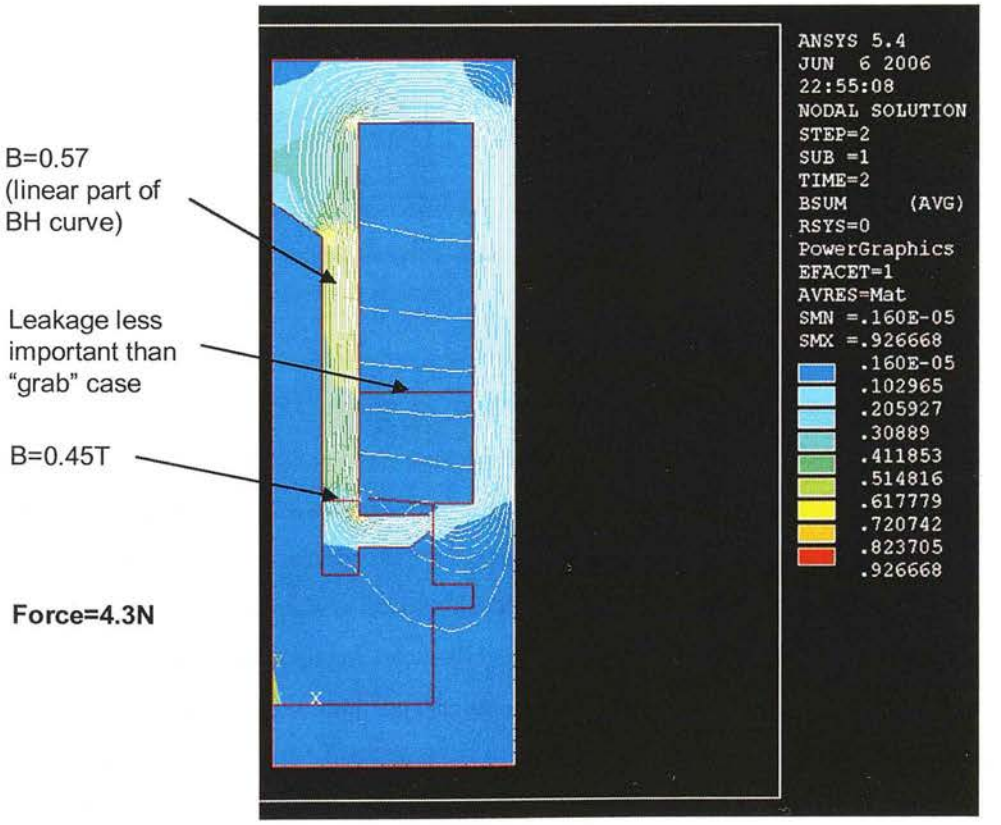


Figure 3-39: Baseline design; "Latch" case

Inspecting the above results, the following conclusions were reached:

- In the "grab" case, saturation in the pole caused most of the flux to leak, bypassing the axial gap and therefore not contributing to useful force. Flux density in the gap was low because most of the flux leaked around the axial gap.
- In the "latch" case, the flux density in the axial gap was low so the latch force was low; however the leakage was much reduced compared to the "grab" case.

To increase the grab force, the saturation in the pole obviously had to be reduced. The first change was to reduce the diameter of the hole in the pole which acts as the guidance for the poppet. In the baseline design, the spring fitted inside the poppet. By making the spring bear on the end of the poppet, the hole diameter could be decreased, increasing the area for the flux in the pole. This meant that the spring could no longer be housed inside the poppet but had to bear on its end, reducing the length of the spring. This could be turned to an advantage, because a shorter spring would have a higher rate. Reducing the poppet diameter from 5.0mm to 4.2mm, increased the grab force to 4.2N.

The other obvious way to increase the area for flux in the pole was to increase its outside diameter. However this directly reduced the area of the coil, and with a fixed maximum current density this would reduce the ampere-turns from the coil available to drive the circuit. Also, as the radial gap between the pole and the outer cartridge shell reduced, the reluctance of this gap would reduce, causing more leakage.

To find the best design, a 10 point sweep was made of the external radius of the pole with the valve in the "grab" configuration, with a constant current density of 25 A/mm^2:

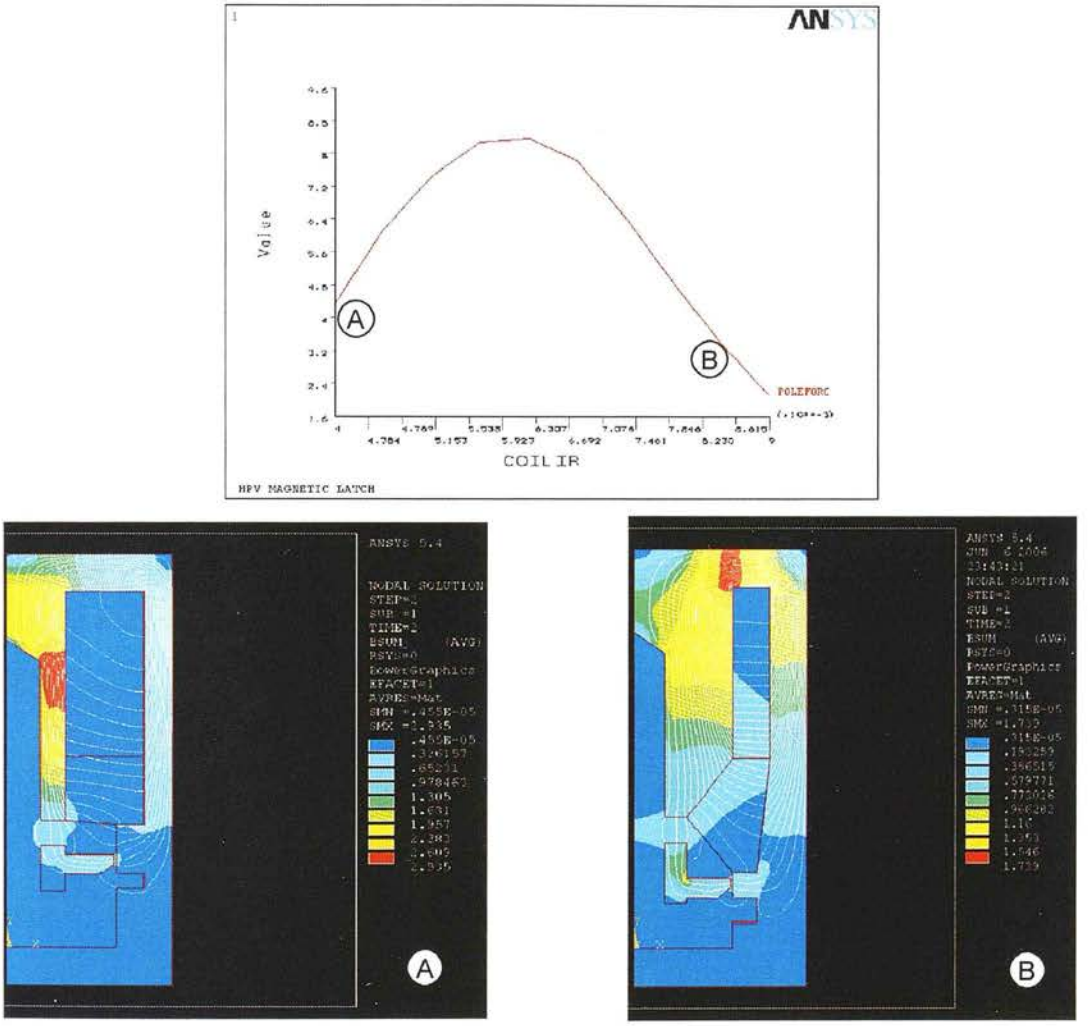


Figure 3-40: Effect of coil internal diameter on grab force;

The graph in the figure above shows the grab force as a function of *coilir*, and plots of flux at the two extremes. The result clearly showed that the best *coilir* (keeping all the other parameters at their starting values) was around 6.0mm; this produced a 195% increase in the grab force, from 4.2N to 8.2N, compared to the starting value. A further sweep with constant ampere-turns from the coil, rather than constant current density, returned a best value of 7.5mm, but this implied increasing current density beyond the established limit of $25A/mm^2$. As this limit depends on the balance between heating power and the coefficient of conductive heat transfer, it was reasonable that the current density limit of the coil could increase as its ratio of surface area to volume increased. The value of 6.5mm was therefore chosen as the basis of further design improvement, with a current density of $28A/mm^2$

The next parameter investigated was *poleer*, the external radius of the pole face. This set the area of the axial gap, where the actuation force was generated.

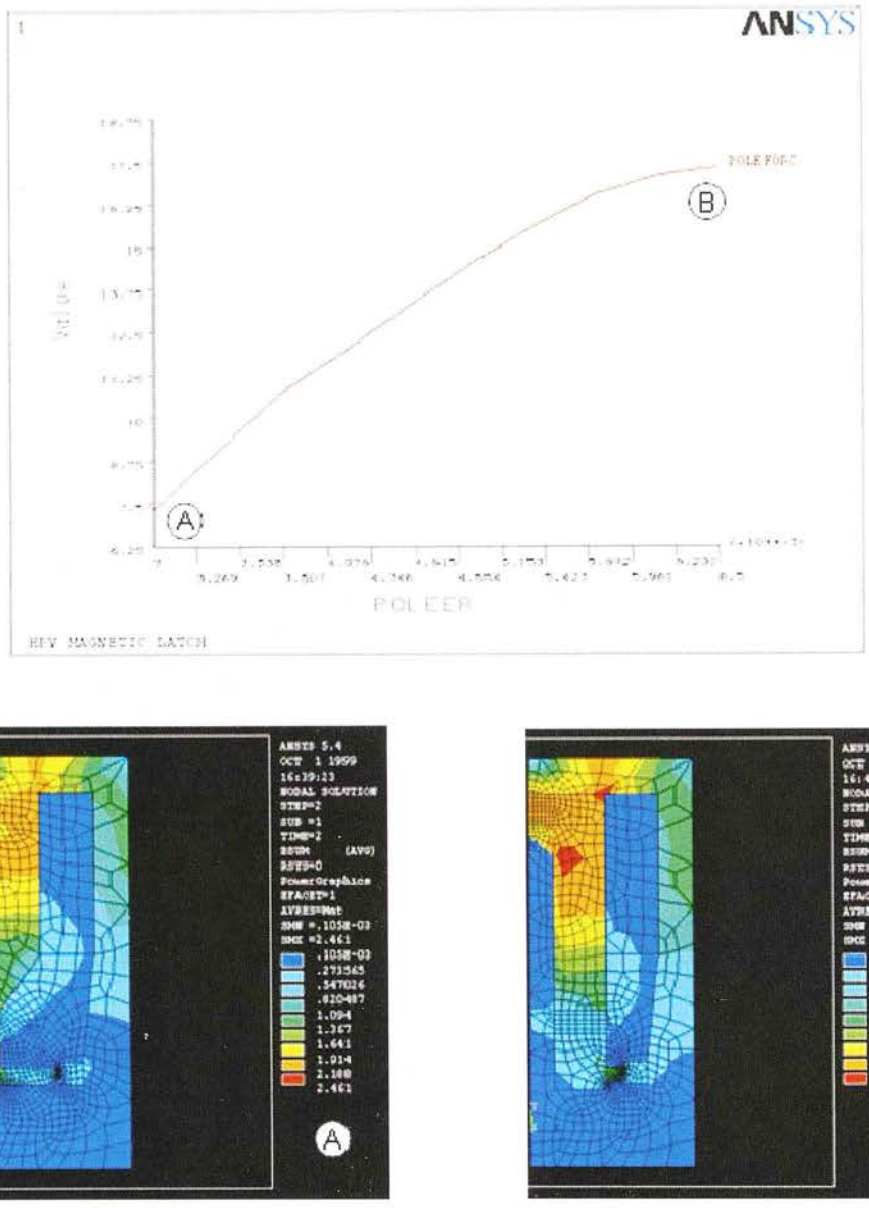


Figure 3-41: Effect of increasing the external diameter of the pole face

It was expected that some degree of flux concentration at the pole would be desirable to generate maximum grab force, but the figure above shows that this is not true. An inescapable side effect of concentrating the flux in the pole is that the reluctance of the axial air gap increases; this caused more flux leakage which counteracted any flux-concentrating effect. In the grab case, the best *poleer* was the same (6.5mm) as *coilir*.

Together, the improvement of the parameters *poleer* and *coilir* resulted in an increase in grab force of 404%, from 4.2N to 17.0N.

Further improvements to the grab force were difficult to find. Changing *ringir* from 7.5mm to 8.0mm gave an increase to 18.2N, but it was not possible to increase this parameter any further because the axial face of the radial gap was used as a sealing face, and this seal was

already as radially thin as possible, being a custom-made PTFE ring rather than a standard o-ring.

It can be seen from the above plots that in order to minimise leakage between the internal and the external radius of the coil, *coiler* should be maximised. If the HPV cartridge was used in free space, this would have the undesirable consequence of increasing flux density in the external return path of the flux in the shell. However in use the cartridge was always inserted into the cavity in the steel pump ring; this can be considered to be part of the flux path, allowing *coiler* to be increased to 21.5mm. This gave a wall thickness of the shell of 1.15mm – judged to be as thin as possible given machining considerations, the requirement to retain sufficient wall thickness for the M24x1.5 thread, and the requirement to transmit the tightening torque of this thread through the shell.

With the grab force much improved, attention turned to improving the latch force. The concept was to design into the pole face a feature which would concentrate flux when the gap was very small, but which did not compromise the grab force.

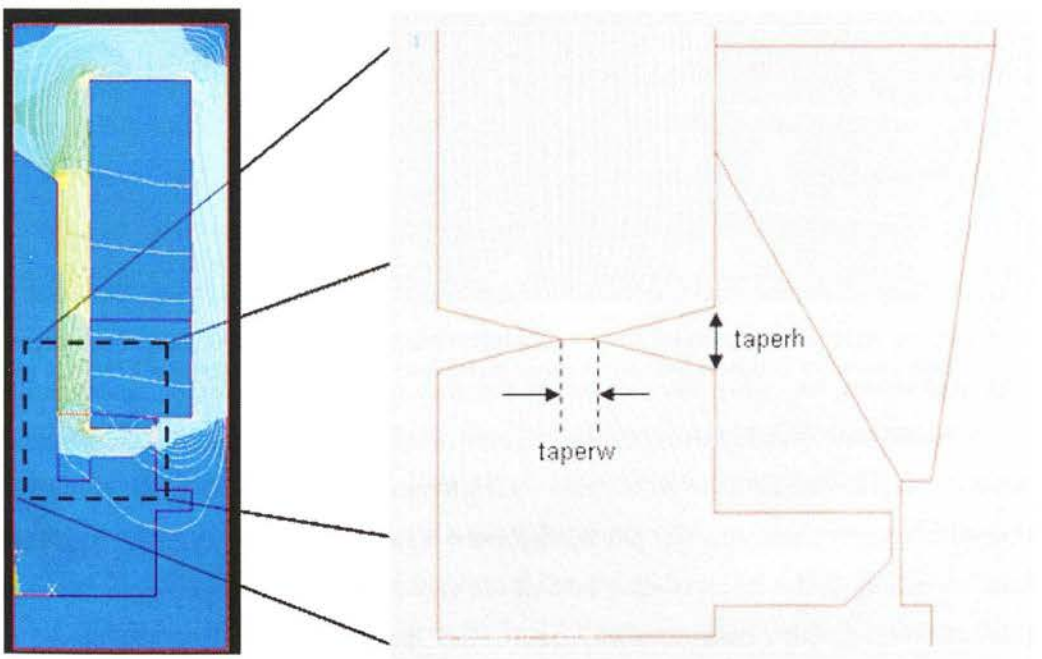


Figure 3-42: Concept of the flux concentrator

The parameters *taperw* and *taperh* were investigated by conducting sweeps for both the latch and grab cases.

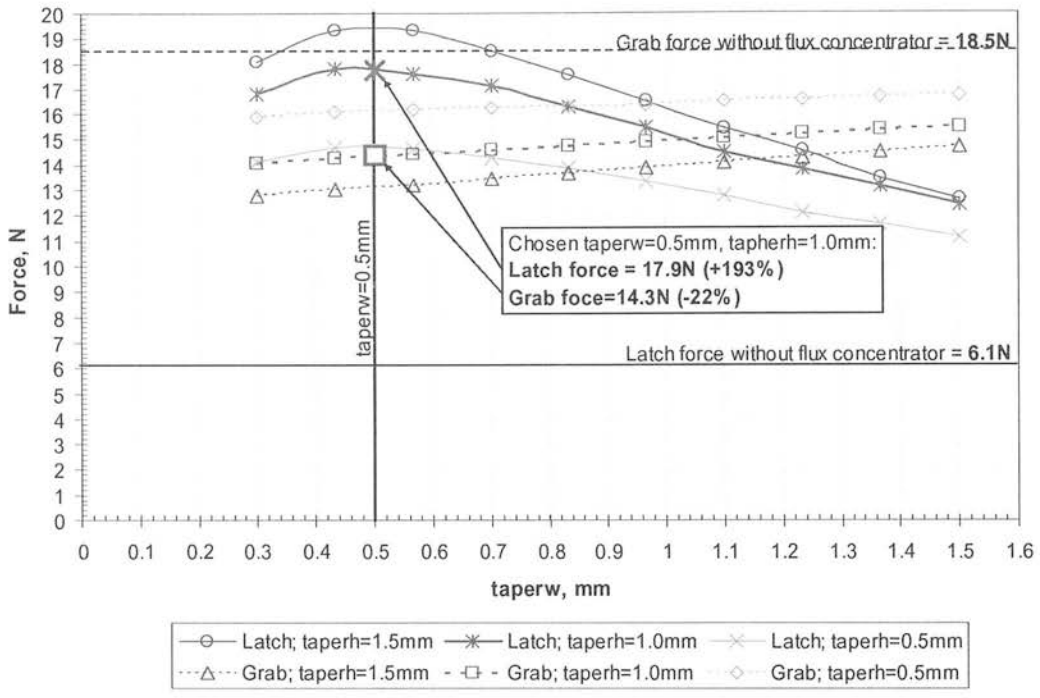


Figure 3-43: Results of parametric design improvement of the flux concentrator

The baseline performance of the latch without concentrator is shown as the two horizontal lines. As *taperw* increased and *taperh* decreased, the flux concentrator feature is disappearing and both the latch and grab forces tend towards the "without concentrator" case. There is clearly a trade-off between latch and grab forces, but the potential improvement to latch force is much larger than the reduction of grab force.

The choice of *taperw* is obvious- the latch force peaks with taperw=0.5mm regardless of taperh, and taperw has fairly weak effect on the grab force. The choice of *taperh* is more finely balanced– the question was, how much is an increase in latch force worth, compared to a reduction in grab force? In the NCSO HPV, latch force determines the strength of the return spring which can be fitted, and this has the dominant effect on t_{tr}. A longer return time causes greater variability of the actual closing time of the valve, and it is the return time rather than the actuation time which determines the most critical timing compromises. The choice was therefore made with a bias towards latch force. The benefit to latch force of increasing *taperh* beyond 1.0mm seems to diminish, while the grab force suffers significantly. Outside the domain of the model, it was apparent that the smaller *taperh*, the more obtuse the taper angle became which had to be machined on to the face of the pole and armature. The more obtuse the angle, the more relative error would be introduced into the actual *taperw* cut by the lathe on the face. In addition, it was thought that a more obtuse angle would exaggerate the squeeze

film damping effect between the faces of the pole and the armature. Taking all these considerations into account, *taperh*=1.0mm seemed a reasonable compromise.

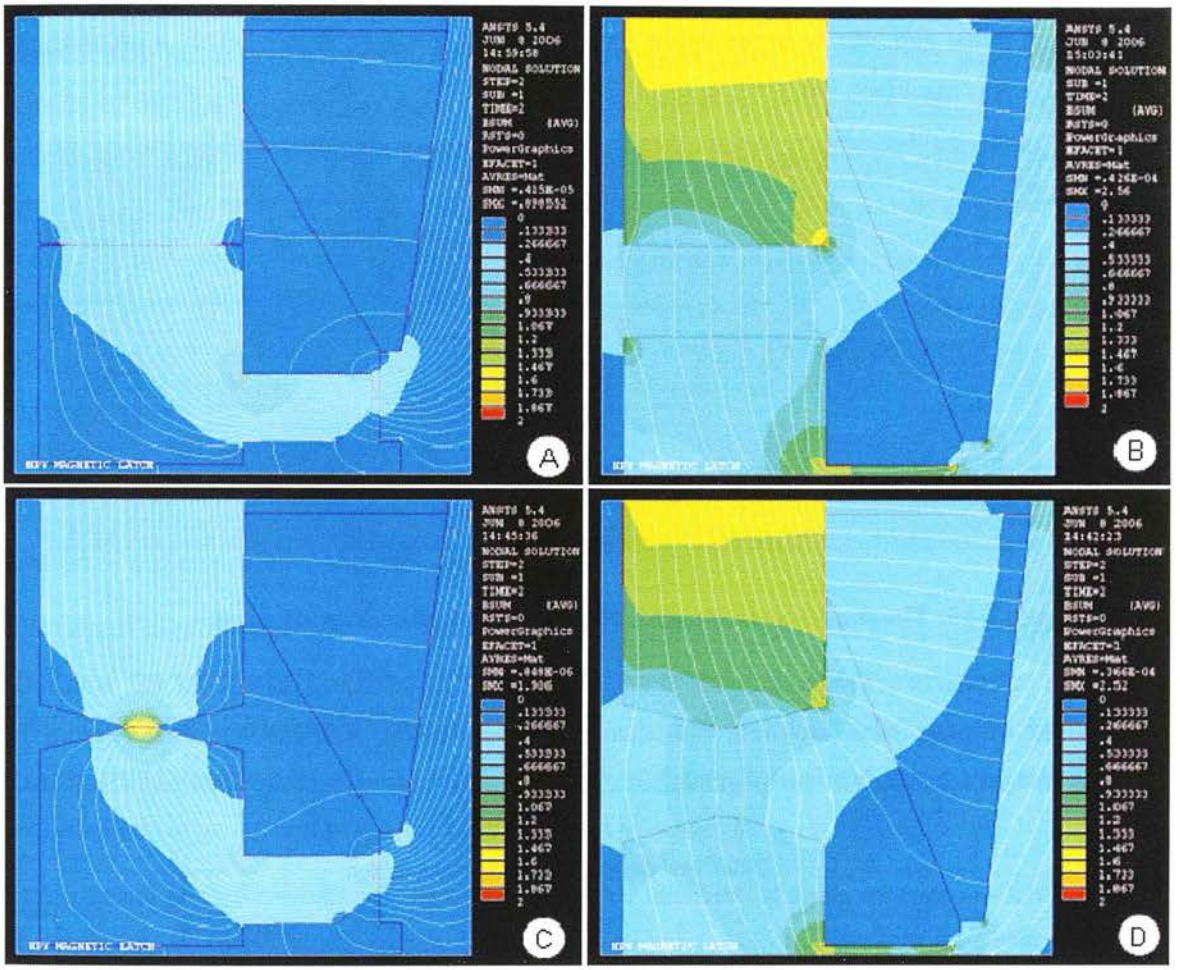


Figure 3-44: Comparing the grab and latch cases with and without concentrator

The above figure shows the effect on the flux distribution and flux density of the flux concentrator; the contour colours of all four plots have the same range (red=2.0T). In the latch case, the flux density in the axial gap is greatly increased with the concentrator (C) compared to without (A). Even though the active area of the gap reduces by a factor of four, the flux density is 4 times higher and therefore B^2 is 16 times higher – leading to a net 4 times improvement in force. In the latch case, the flux density near the edges of the gap is reduced with the concentrator (D) compared to without (B), while the flux density in the middle is similar– the "concentrator" has the effect of encouraging the flux to diverge in the grab case.

With the grab and latch force improved, attention turned to the moving mass – a major determining factor in fast valve response. As the armature must be ferromagnetic, its density was much higher than the thermoplastic poppet and therefore minimising the volume of the armature was the highest priority.

The axial length of the armature was reduced as much as possible, the armature tending towards a disk shape. While attention was focussed on this part, it was apparent that the reluctance of the radial air gap could be reduced by removing the chamfer on the poppet side of the radial gap, visible in the figure above. This led to the geometry below, a prototype of which was manufactured:

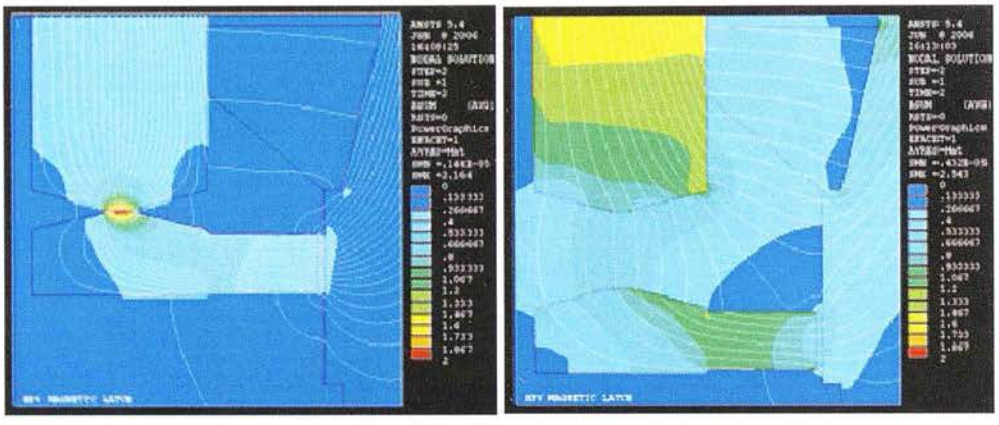


Figure 3-45: The final armature geometry with minimised moving mass

The increase in the radial gap area increased the latch force from 17.9N to 22.4N, with little effect on grab force. This is to be expected, because in the latch case the radial gap is the dominant reluctance in the magnetic circuit, whereas in the grab case the axial gap dominates.

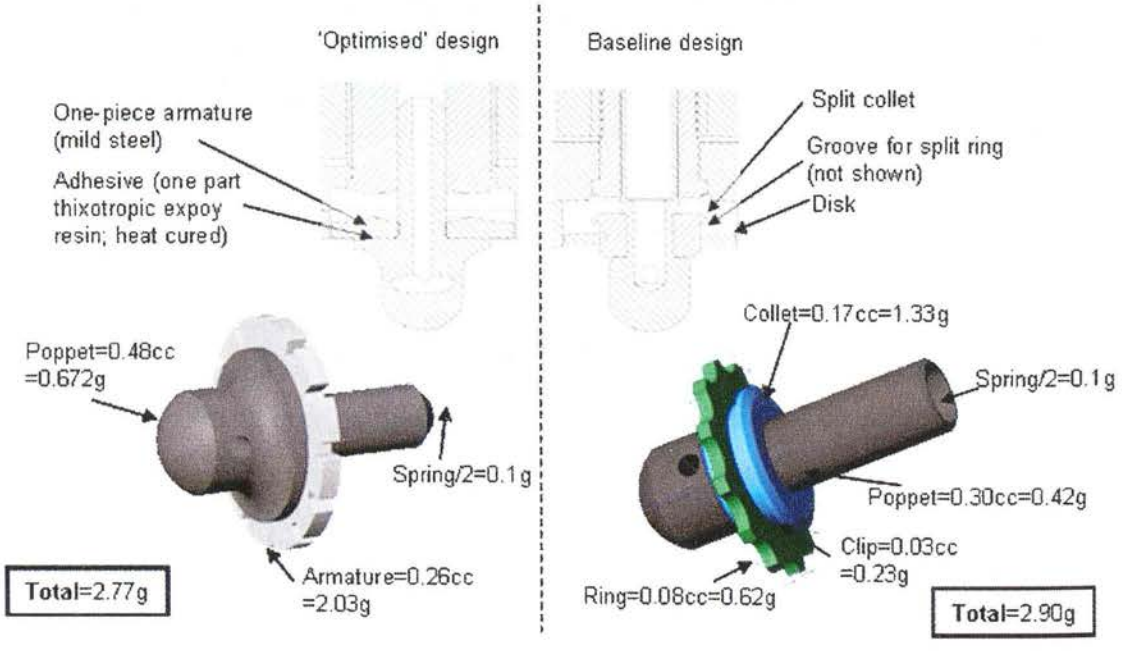


Figure 3-46: Redesign of the poppet and armature of the improved HPV

When the armature was re-designed, the attachment method between armature and poppet also had to be revised because there was no longer axial room for the mechanical attachment

arrangement designed by Rampen. An adhesive bond seemed to offer a low space and weight but there were doubts about the strength of bond achievable. The poppet interface surface was maximised by flaring out the poppet at the interface, yielding 90mm^2 of bond area. The manufacturer of the poppet material recommended an epoxy adhesive and gave a guideline of 5.9N/mm2 as an achievable lap shear strength; if this could be achieved in tension then the bond would fail at 531N, at least 4 times the maximum solenoid force. Tests were performed with prototype parts of the candidate geometry to verify this; two samples failed at >850N tensile force. The environment of hot oil is not ideal for adhesives, but in practise it proved to be reliable as long as the surfaces were carefully roughened and de-greased before being bonded in a jig.

3.4.9 Test results for the improved HPV

The current required to hold the improved prototype HPV closed against its return spring was measured, and compared to the FEA prediction.

The improved prototype took 77 ampere-turns (180mA at 432 turns) from the coil to hold the 14.5N of the return spring. The ANSYS model predicted 18.6N at 77 ampere-turns; the difference could be explained by machining tolerances of the width of the flux concentrator and the radial air gap, and non-axisymmetric slits which were introduced in the armature at the radial gap in an attempt to reduce the effect of fluid damping.

Experimental latch force data for the baseline and the improved designs was not available at the same ampere-turns, making direct experimental comparison of the improvement difficult. However, they can be compared by making the assumption that the solenoid is working in the linear region, in which case eq. 41 (p.99) applies and the force is proportional to the square of the ampere-turns. The graph below shows the data points of latch force vs. ampere-turns achieved by the baseline and the improved designs. Also shown are curves showing the ideal relationship between ampere-turns and force, according to: $F=k\cdot AT^2$, where k adopts eight discrete values (0.002,0.004,0.006…0.016).

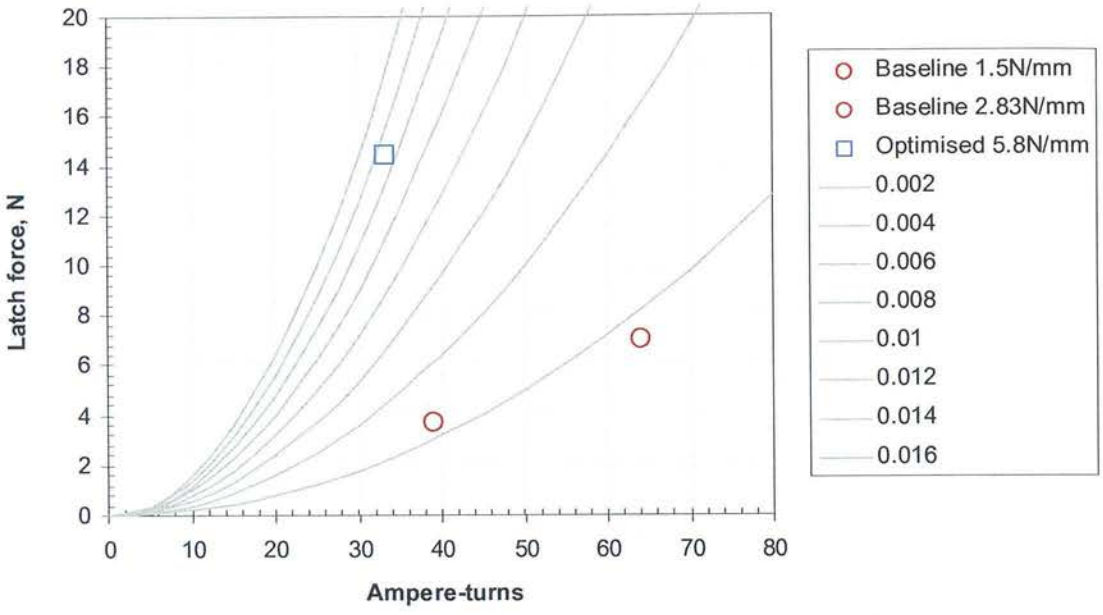


Figure 3-47: Comparing latch force of baseline and improved HPV

The two data points for the baseline results lie close to the k=0.002 curves, while the improved design point lies between the k=0.010 and k=0.012 curves. From this it can be estimated that, for a given ampere-turns, the improved HPV gave between 5 and 6 times the latch force compared to the baseline design. This is consistent with the FEA results which predicted that latch force should be increased from 4.3N to 22.4N, a factor of improvement of 5.2.

The time delays for the improved prototype HPV were measured; these results are shown below compared to the baseline design.

Criterion	Original design	Improved design
FEA "latch" force	4.3N	22.4N
FEA "grab" force	2.1N	14.2N
Moving mass	2.90g	2.77g
Spring	1.5N/mm; 0.5mm preload. Force (latched) = 3.75N Force (seated) = 0.75N	5.8N/mm; 0.5mm preload. Force (latched) = 14.5N Force (seated) = 2.8N

t_{da}	2.3	3.0
t_{ta}	3.0	2.1
t_{dr}	2.0	2.0
t_{tr}	2.70	1.2

Compared to the baseline design, the improved HPV had a longer actuation delay; this can be attributed to the much higher spring preload and rate. However the actuation transit time was reduced, and the total actuation delay was slightly less (baseline=5.3ms, improved =5.1ms). The return delay time was the same for both designs. For the DDPM performance it has been noted in section 3.2.6 that the return transit time has the most significant effect on the overall DDPM performance; this reduced from 2.70ms to 1.20ms. This can be attributed to the factor of 3.9 improvement (3.75N to 14.5N) of the spring force which acts to return the poppet to its seat.

The ultimate measure of the improvement of the HPV design is in the performance achieved by the DDPM. In operation it was seen that the de-pressurisation time was much more consistent, the variability of the HPV closing time reducing from +/- 1.5ms to +/- 0.5ms. This meant that the *HS* time could be set later without risk of *HP idle* cycles occurring due to insufficient de-pressurisation time. With the baseline design, the maximum displacement that could be sustained at 1500rpm, 100 bar was 75% of the geometric displacement. With the improved design, this improved to 88%.

3.4.10 Discussion of the effect of improving valve timing accuracy

The significance of the improvement achieved in displacement to the viability of the DDPM as a propel motor depends on how the motor is used. In a hydrostatic transmission with a wide ratio spread (such as a wheel loader), when propelling the vehicle, the motor does not operate at full displacement at full speed, because there is insufficient pump flow; the motor may only need to operate at 25% displacement at full speed. However in a secondary-controlled application there may be enough accumulator volume to allow the motor flow to exceed the pump flow for a short period. In both cases, the flow when braking (when the motor operates as a pump) is not limited by the pump capacity, and hence full displacement at full speed may be a legitimate operating point; the flow capacity of the hydraulic circuit may be the limiting factor here.

For best power density, a propel motor should be geared to spin at maximum shaft speed at the maximum speed of the vehicle. The maximum speed of the DDP propel motor described

in the next chapter was 1800rpm, but this was limited more by controller computational power than by the electro-mechanical fundamentals of solenoid commutation. Since this work was done, a DDPM at this scale has been made by AIP which achieved 85% of the geometric displacement at 3500rpm, and it is expected that a similar performance will be achieved at 3000rpm by machines of larger scale, suitable for mainstream propel application.

At low speed, the effect of the variability of the valve return time becomes less significant in the context of a longer overall cycle time; this means that as speed falls, the DDPM should approach closer to displacing 100% of its geometric displacement, although at very low speed leakage will become significant.

It is clear that to achieve the best power density, the aim should be that the maximum speed of the DDPM should be limited not by the commutating valve delays, but by the more fundamental considerations of cylinder breathing pressure drop and piston pad bearing power losses. Therefore the significance of improving valve timing accuracy may be mainly in its effect of increasing the maximum operating speed of the DDPM towards these fundamental limits.

3.5 Transient analysis of the HPV

So far, the analysis presented of the DDPM solenoids has been quasi-static. Rampen (1992) noted that it would be desirable to develop a transient model of the valve solenoids, to understand more fully the factors affecting the delay times which are crucial to accurate commutation. This section presents such a model and compares it with experimental results.

3.5.1 Literature review

Aldefeld (1978) investigated transient effects in solenoids using a digital computer. He used a 2D time-stepping finite-difference method to simulate transient excitation in an axisymmetric solenoid. Although he achieved good results with his chosen problem case, the method suffers from the difficulty of accurately modelling fine geometry with a fixed grid of points. The finite element method which ultimately achieved dominance allows the geometry to be more accurately modelled (Kawase et al. 1991).

Some researchers have considered that accurate simulation of the time-domain response of solenoid-actuated valves depends on solution of the 2D axisymmetric time-varying magnetic field problem with FEA, directly coupled to 1D models of the mechanics and the non-linear electrical circuit. Applications on automobiles have stimulated much research in this area

Pawlak et al. (1988) made such a model to study the problem of automotive solenoid valves with pulse width modulated (PWM) electrical excitation, with good matching to experimental results. Sangha et al. (1994) made a similar model which also compared well to observations.

Kawase et al. (1991) used a similar model to investigate the effect of different return spring rates on the response time. Not surprisingly, they found that a weak spring favoured a low actuation time, while a strong spring favoured a low return time - they defined the optimum spring rate as that which the sum of both times was minimised. Ohdachi et al. (1991) considered the effect of minimising moving mass on the response time.

The coupled FEA/1D formulation seems to offer the most general and powerful method of simulating the motion of solenoid-actuated valves. However the usefulness of the approach suffers from the fact that at every time step of the 1D models, the FEA model has to be re-meshed, solved and the results communicated to the 1D model. This means that it takes many (perhaps hundreds) of FEA solutions to get one simulated valve motion solution. This is the case even if it is the effect of the 1D parameters (such as excitation voltage or spring rate) that are the focus of the investigation rather than the valve geometry. From the author's experience, each FEA solution takes around 30 seconds to perform for realistic geometry complexity at acceptable mesh density, with a 1Ghz Pentium III PC. Therefore hours may be

needed to analyse each design. This drawback has been noted by others (Lequesne 1990, Chladny et al. 2005).

The increased interest in fast-acting solenoid valves in the fluid power industry has also stimulated much research. The approach of most of these researchers is different from those from the automotive industry. Often the valve under consideration is just one component of a wider hydraulic system, and it is the behaviour of this system that is of primary interest, not the detailed response of the valve itself. This has led to wide adoption of methods which characterise the solenoid with approximations which are much faster to solve.

Pohl et al. (2001) present a simple lumped-parameter (LP) model that does not take account of saturation effects, because "such operating points are normally never reached in these applications". This assumption may be valid for solenoids designed for continuous proportional operation, such as exhaust gas recirculation valves in diesel engines, but is not valid for the over-driven digital solenoid valves described in this thesis, which require the maximum force from an actuator of limited volume for a short period of time, and hence tend to push the magnetic circuit into the non-linear region.

Sethson et. all (1993) used an analytical LP model for the solenoid, using iteration to deal with magnetic saturation of the reluctance network - similar to the method described by Roters. No FEA model was involved. The authors acknowledged that the results are only valid for cases where the magnetic circuit reluctances do not change significantly with armature position. The model assumes the ideal inverse-square relationship between armature position and force, and does not deal with problems where flux leakage paths change significantly as a function of current or armature position. A fairly good match is achieved with experimental results from a small normally-open poppet valve, although this solenoid is specified for continuous duty, so is not driven as hard as a solenoid in a DDPM must be.

Lequesne (1990) describes a more complex LP model, with saturation taken into account, and presents favourable comparisons with time-domain coupled FEA/1D analysis.

Cheung et al. (1993) obtained their 1D solenoid model by curve-fitting experimental data, and present a qualitative match between the output of the model and experimental results. Zavarehi et al. (1999) followed a similar approach but with a much more complex fluid side of the model.

Wang (1993) used magnetostatic FEA to derive maps of force and inductance for the solenoid under investigation. These maps were then used as lookup tables in the 1D motion and electric circuit models. A good match with experimental results was achieved for a valve with 2ms transit time.

Chladny et al. (2005) compared two methods of implementing an LP model. In one, the results from many magnetostatic FEA solutions over a range of armature positions and currents were used to generate inductance and force lookup tables similar to Wang (1993). These were used to calculate force and *di/dt* for the lumped-parameter model; they call this a "hybrid FEA/LP model". In the other method, the FEA results are used to generate parameters for an ideal inverse-square law model with no saturation. The latter model did not match experimental results well when the armature was close to the pole – here, the flux density in the steel parts went into the non-linear region. The authors conclude that the hybrid FEA/LP model offers the best balance of accuracy and computational efficiency; for this reason, this method was selected by the author for this work.

Recently, integrated software packages for multi-domain 1-D system simulation have achieved commercial success. Amesim (2005) offers an interface to the magnetic FEA program "Flux2D" via lookup tables in the same way as Wang(1993) and Chladny (2005). Using these tools the analyst can import tables created with Flux2D into a one-dimensional solenoid object, which can then be connected to an electrical circuit and a one-dimensional mechanical system.

A truly complete model of the transient effects should consider eddy currents – these are transient parasitic currents which flow in the steel parts of the solenoid due to the rate-of-change of flux; effectively the steel parts of the circuit form a shorted-turn which is coupled magnetically to the coil, like a transformer. The effect is to damp the rate-of-change of the flux in the circuit, causing an increase in the time constant of the coil. This effect is typically included in transient FEA analyses e.g. Pawlak et al. (1988) and Sangha et al. (1994). However it is often ignored in LP models e.g. Chladny et al.(2005), Zavarehi et al. (1999) and Cheung et al. (1993), on the basis that it has only a marginal effect in many solenoids.

To assess whether this was the case with the DDPM HPV, a prototype of the improved HPV was slit radially such that the electrical continuity of the magnetic circuit was interrupted. This time constant and total actuation delay for this solenoid were indistinguishable (i.e. less than 2%) from the intact solenoid in the experiments. For this reason eddy currents were not included in the author's model.

3.5.2 A hybrid FEA/LP transient solenoid model

This section presents a hybrid FEA/LP model similar to those by Chladny et al. (2005) and Wang (1993).

The inductance of a coil is defined as (Roters 1945):

$$L = \frac{\lambda}{i} = \frac{N \cdot \varphi}{i} \tag{42}$$

where λ is the flux linkage of the coil in the magnetic circuit, N is the number of turns of the coil and ϕ is the total flux in the circuit.

Assuming perfect linearity, an expression for the inductance of the coil in the plunger solenoid can be derived (Roters 1945, Chladny 2005):

$$L(x) = \frac{2 \cdot B}{k - x} \tag{43}$$

Assuming that the coil can be modelled electrically as a resistor R_c in series with an ideal inductor L , the current in the coil follows the equation (Chladny 2005):

$$V_L = i \cdot R_C + L \cdot \frac{di}{dt} \tag{44}$$

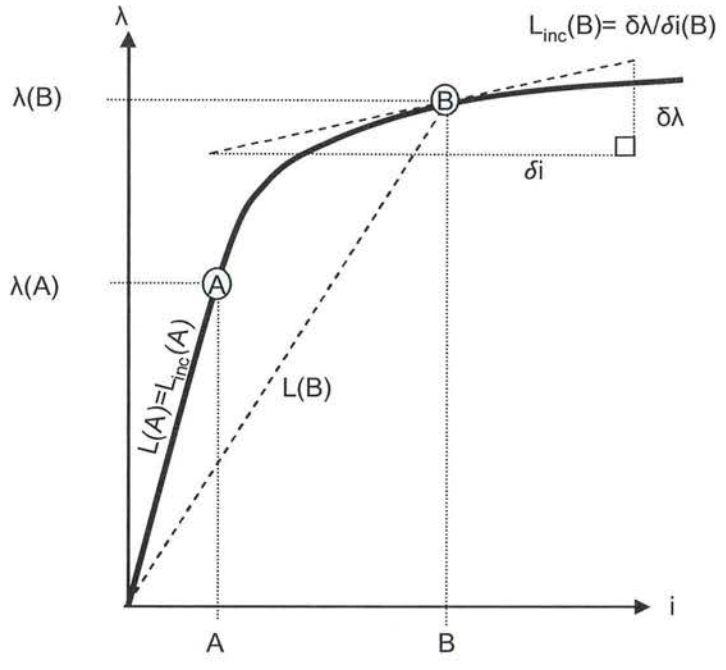
The solution of this first order differential equation for a step change in voltage is the equation of current rise in a *RL* circuit, where $\tau = L/R_C$ is the time constant of the coil:

$$I(t) = \frac{V}{R_c} \cdot (1 - e^{-\frac{t}{\tau}}) \tag{45}$$

However these equations are only valid under the following assumptions:

- the permeability of the materials in the circuit is constant;
- the velocity of the armature is negligible.

Both of these assumptions are incorrect for high-speed, over-driven solenoids. Dealing first of all with the changing permeability of the material, it is useful to introduce the concept of the *incremental inductance*, the rate of change of flux linkage with respect to current at the present armature position and current:



3-48: Distinguishing between inductance L and incremental inductance L_{inc}

$$L_{inc}(i) = \frac{\delta\lambda(x,i)}{\delta i} \tag{46}$$

The incremental inductance is a function of both armature position and coil current so is best visualised in the form of a surface plot. With a parametric FEA model, a lookup table of incremental inductance can be generated by solving the magnetostatic problem over a range of both x and i.

The simple linear equation of current in an RL circuit can be generalised as (Cheung et al. 1993):

$$V_C = i \cdot R_C + \frac{d\lambda}{dt} \tag{47}$$

Where:

$$\frac{d\lambda}{dt} = \frac{\delta\lambda(x,i)}{\delta i} \cdot \frac{di}{dt} + \frac{\delta\lambda(x,i)}{\delta x} \cdot \frac{dx}{dt} \tag{48}$$

The first term in the sum above is the inductive e.m.f. and the second term is the motional e.m.f.

The motional e.m.f. can be derived from the force exerted on the armature by (Cheung et al. 1993)

$$\frac{\delta\lambda(x,i)}{\delta x} = \frac{F}{i} \tag{49}$$

Therefore the motional e.m.f. is

$$V_{motion} = \frac{F(x,i)}{i} \cdot \frac{dx}{dt} \tag{50}$$

and the dynamics of the coil simplify to:

$$V_C = i \cdot R_C + L_{inc}(x,i) \cdot \frac{di}{dt} + \frac{F(x,i)}{i} \cdot \frac{dx}{dt} \tag{51}$$

Neglecting flow forces, the equation of motion of the armature between the end-stops is:

$$F(x,i) = m \cdot \frac{d^2x}{dt} + k_{spring} \cdot (x + preload) + k_{damp} \cdot \frac{dx}{dt} \tag{52}$$

Where m is calculated from measurements or by calculating volumes in the solid model,

k_{spring} is the spring rate specified by the supplier (or easily measured),

preload is the compression of the spring in the fully returned position (note that if *preload* is negative, the term *x+preload* in the above equation must be limited to be >=0 if it is assumed that a compression spring is used, which cannot provide a tension force),

$L_{inc}(x,i)$ and $F(x,i)$ are look-up tables that are generated from the FEA model.

Of the mechanical parameters, k_{damp} is the most difficult to characterise. It is likely to be a function of x due to squeeze film effects as surface approach close contact e.g. in the flux concentrator (Rampen 1992 p.122). For the purposes of this model it is assumed that function takes the following simplified form:

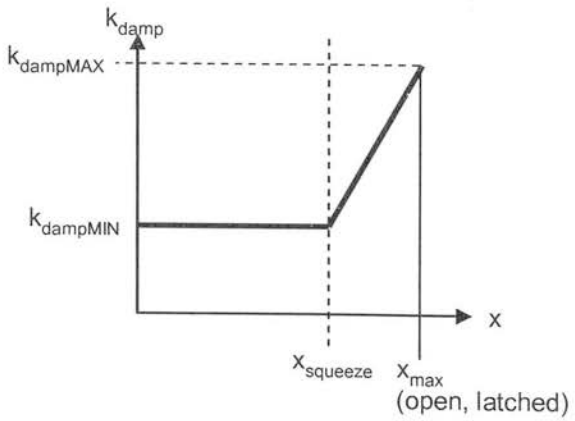


Figure 3-49: Simplified squeeze-film damping

3.5.3 Modelling method

An ANSYS script generated 2D maps of force $F(x,i)$ and flux linkage $L_{inc}(x,i)$ for the final improved HPV design. These maps were interpolated and re-sampled with a parabolic curve-fit, from the public-domain XLXtrFun library (Rauch 1999), to make lookup tables with 40 entries in each dimension, giving a total of 1600 entries. At each time-step these tables were interpolated with a parabolic surface fit routine from the same source called *InterpMatrix*.

A 1D mechanical model was constructed in Microsoft Excel 97 according to eq. 52 with the effects of mass, spring, solenoid force and damping – both linear and simple squeeze film. This was coupled to a lumped-parameter electrical circuit model according to eq. 51, where the incremental inductance of the coil was calculated by a lookup of the table of $(d\lambda/di)$ as a function of position x and current i from the FEA model. The motional e.m.f. was calculated from the velocity at the present time-step and the table of dF/dx – derived by differentiating the force map with respect to x. The effect of pulsing (PWM) was accounted for by ratiometric treatment of the exciting voltage and the flywheel resistor and diode volt drops. The high-frequency saw-tooth form of the current during pulsing was not simulated, because this would have dramatically increased the number of time-steps required, and therefore slowed down the simulation. A fixed time-step of 0.01ms was used; solving for 30ms of motion took around 20 seconds with a 1GHz Pentium III PC.

Maps from the FEA of the HPV

The maps derived from FEA analysis of the improved HPV design are presented below.

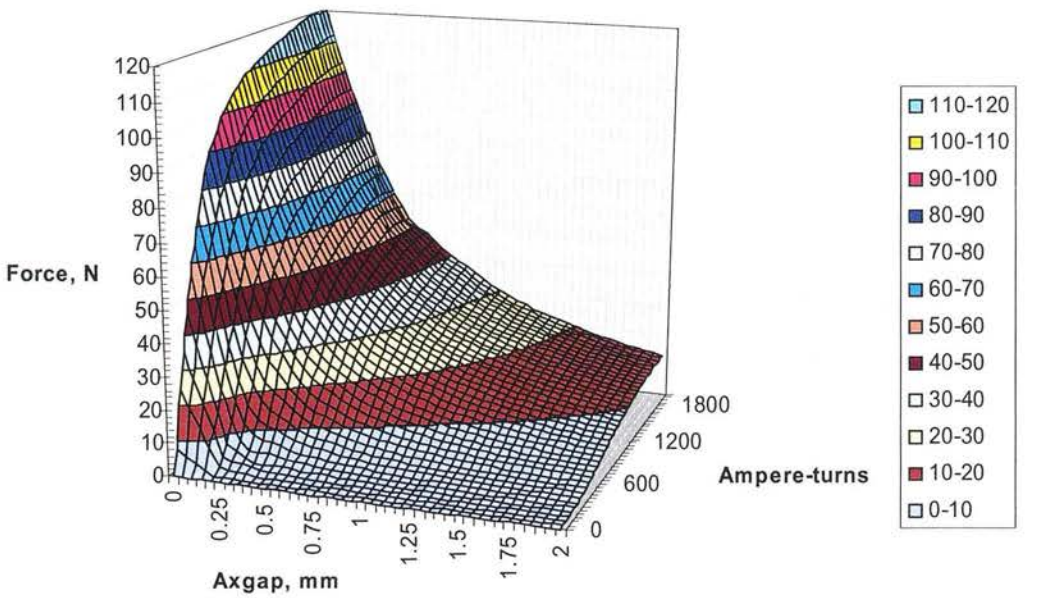


Figure 3-50: Map of force vs. position and ampere-turns from FEA of the HPV

The map above shows clearly the effect of saturation in the magnetic circuit. Using the linear analysis of eq. 41 (p.99), it would be expected that at a given axial gap, the force should be proportional to the square of the current. This is seen more clearly in the graph below which shows the force vs. ampere-turns for 0mm, 1mm and 2mm axial gap.

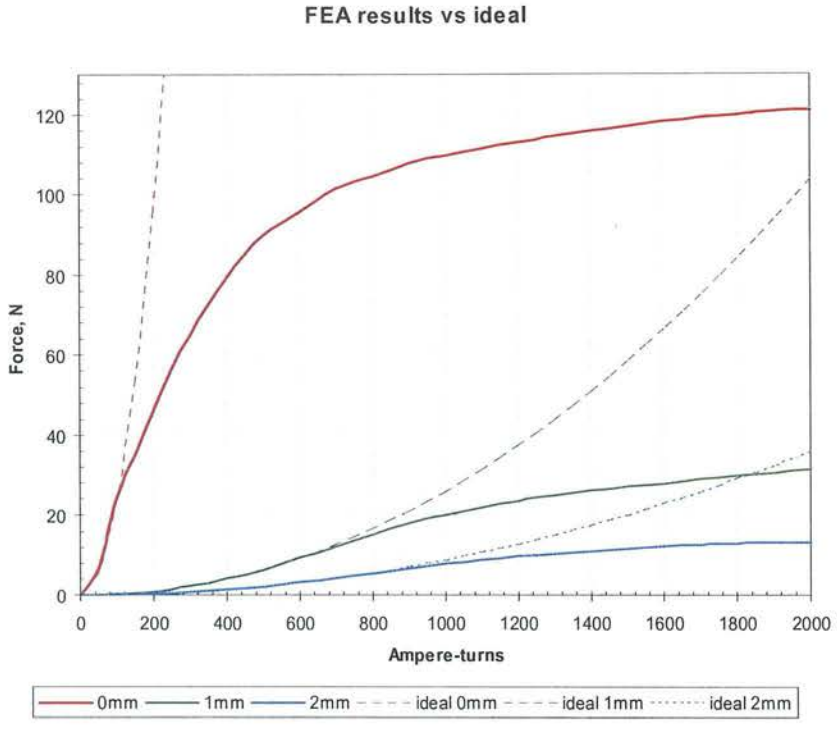


Figure 3-51: Solenoid force vs. ampere-turns in three armature positions; comparison of FEA results with the ideal model

The dotted lines above are of the form $F=k\cdot(AT)^2$ as predicted by eq. 41 (p.99). It can be seen that at maximum axial gap, the ideal equation is useful up to around 800AT, while at minimum axial gap it is useful only up to 100AT. Clearly a linear approximation is inadequate for the HPV solenoid and this confirms the choice of the hybrid FEA/LP modelling approach.

The map below shows the total flux as a function of axial gap and ampere-turns.

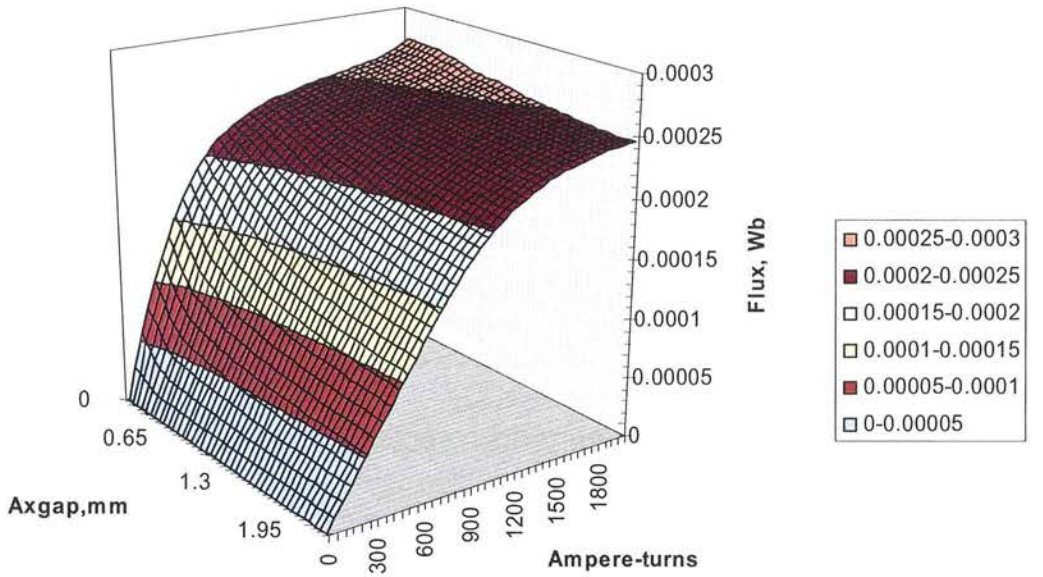


Figure 3-52: Flux vs. position and ampere-turns from FEA

The effect of saturation is clear in this map as the pronounced flattening at high ampere-turns and small gaps. With the linear model, flux is proportional to ampere-turns; with maximum gap this is useful valid up to around 800AT, while at zero gap it is useful only to around 100AT.

This gradient of this map *dAT* is shown below; this is the incremental inductance of a single-turn coil placed in this magnetic circuit.

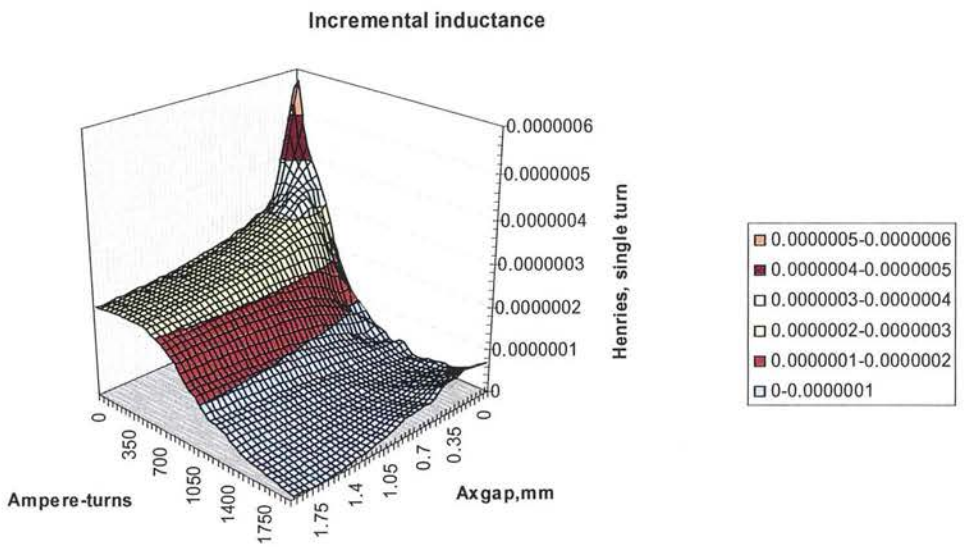


Figure 3-53: Incremental inductance; the gradient *dAT* of above

Clearly at low ampere-turns and large axial gap the incremental inductance is reasonably constant and therefore the coil in this region would display ideal first-order time constant response to a step voltage excitation. However, as ampere-turns increases the incremental inductance drops dramatically. Also evident is the higher incremental inductance at low ampere-turns and zero gap ("latch" condition); here the reluctance of the circuit is at a minimum. In this position it takes only a small amount of ampere-turns to saturate the steel in the flux concentrator, as shown by the dramatic reduction of incremental inductance as ampere-turns rise.

3.5.4 Experimental results for comparison with the model

To compare with the simulation results, the actuation behaviour of the HPV and LPV solenoids for the propel DDPM was investigated by experiment. To minimise the effect of fluid damping, which is difficult to characterise for subsequent modelling, the experiments were conducted in air (with the solenoids installed in the steel pump ring cavity); when flooded with oil the actuation time was typically 10% higher. The drive circuit was simple low-side FET switch supplied with a 24.5ms gate pulse from a pulse generator; there was a flywheel diode (forward voltage 0.6V) but no flywheel resistor. A Hall-effect current transducer was placed in series with the coil. The current trace was recorded with a digital storage oscilloscope at a range of excitation voltages. The model was only implemented for the HPV solenoid but experimental results are also shown for the LPV for comparison.

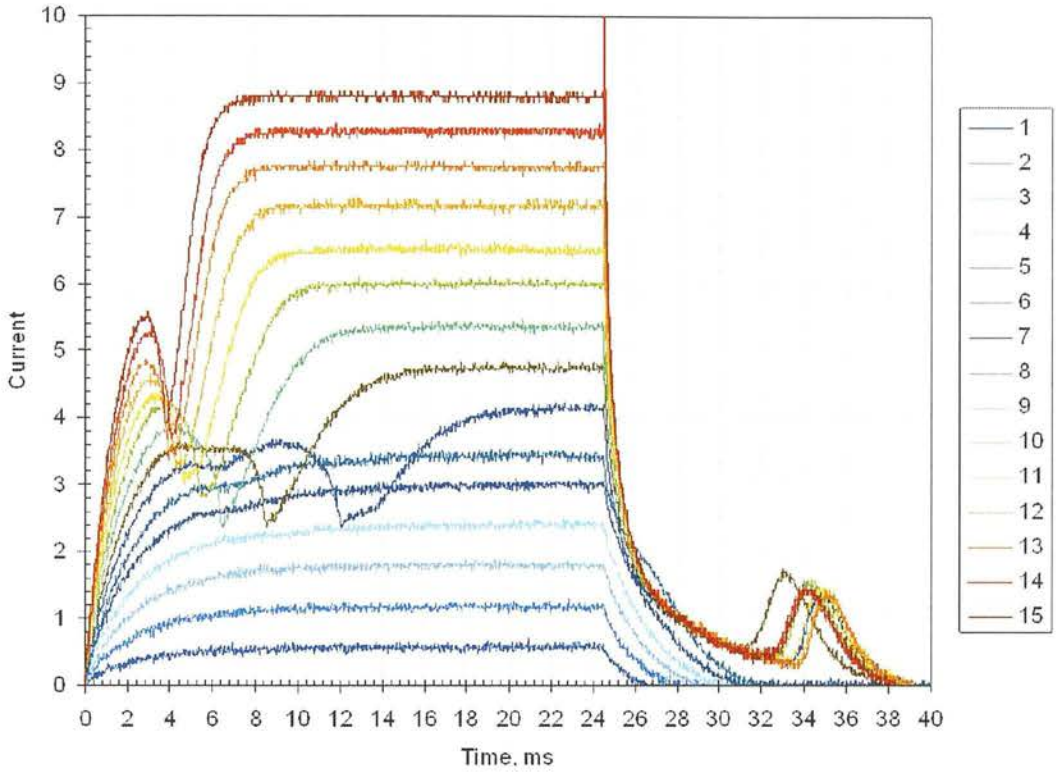


Figure 3-54: Step response of coil current vs. time for the LPV at a range of drive voltages

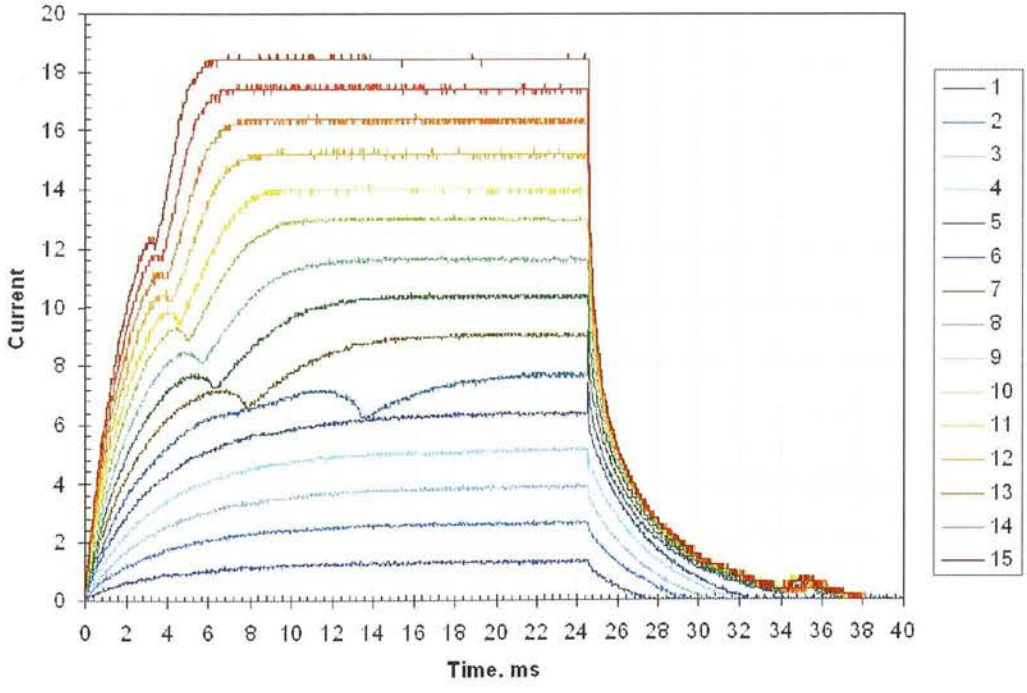


Figure 3-55: Step response of coil current vs. time for the HPV at a range of drive voltages

For both solenoids, at the lower voltages, the current can be seen to rise initially in an ideal first-order form. There is a critical drive voltage below which both solenoids do not actuate, because insufficient force is generated to overcome the spring pre-load. As the armature starts to move the current curves downwards and there is a clear inflection when the armature hits the pole; this is caused by motional e.m.f. (proportional to the product of force and velocity) suddenly reducing from maximum to zero. At the higher voltages the current more rapidly levels off to $i=V/R$ because the incremental inductance reduces as the magnetic circuit saturates.

At t=24.5 the drive voltage steps to zero and the current decays, recirculating though the flywheel diode. At the higher voltages the initial decay is sharper because of the low incremental inductance at high ampere-turns and zero gap. When the solenoid force is less than the spring force, the armature starts to move toward the default position, the motional e.m.f. generated in the process causing a transient increase in current. This is more pronounced in the LPV than the HPV, because the return spring is much stiffer.

3.5.5 Comparison of the model with experiment

The HPV was modelled using the maps above as look-up tables, and the following 1D parameters:

Parameter	Value	Source
Moving mass	2.83g	Measurement
Spring rate	5.8N/mm	Manufacturer's data
Spring pre-load	0.5mm	Measurement
Damping	0 Nms^{-1}	Assumption
Coil turns	180	Measurement
Coil resistance	0.77 ohms	Measurement

Figure 3-56: Table of HPV parameters for the transient model

For comparison, the above results from the experiment on the HPV are shown below with more detail of the actuation event.

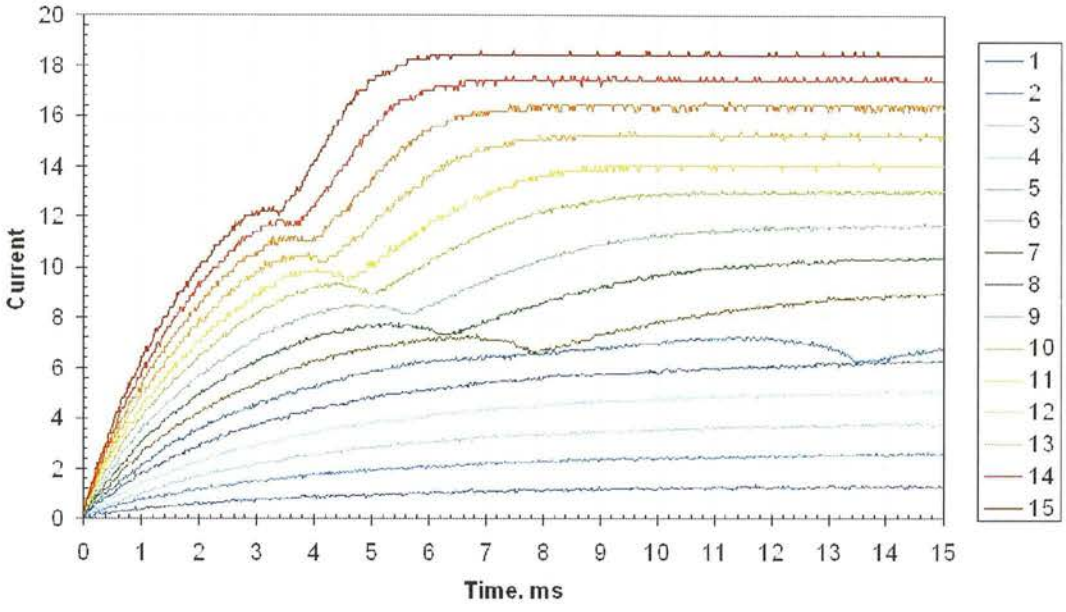


Figure 3-57: Experimental results of current vs. time for the HPV, subject to transient excitation at a range of coil voltages.

The results from the model are shown below.

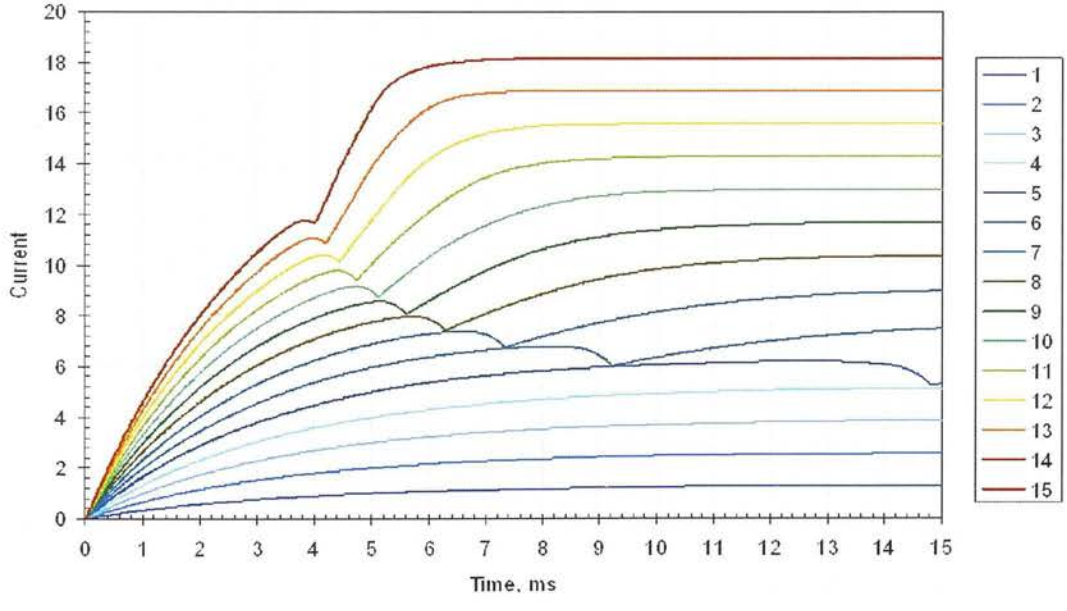


Figure 3-58: Output of the hybrid FEA/LP model of the HPV, subject to transient excitation at a range of coil voltages.

The model and experimental results are compared on the graph below (only every second voltage is shown for clarity).

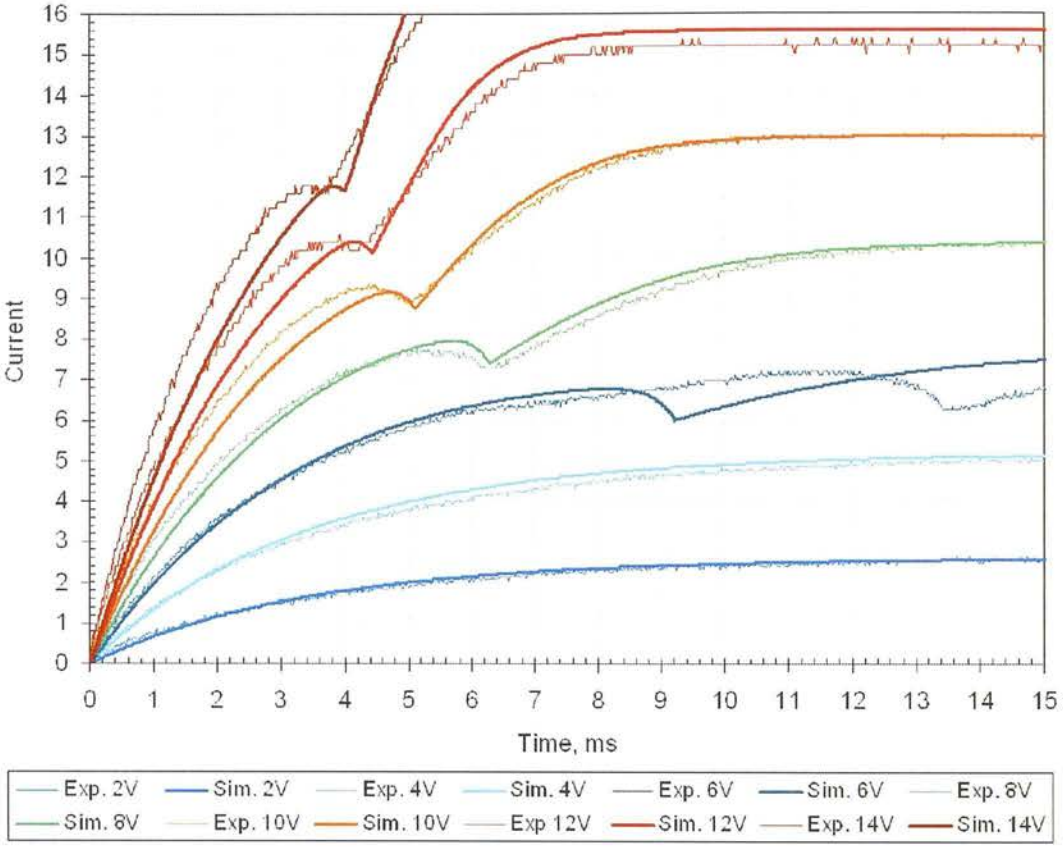


Figure 3-59: Comparison of the experimental results and the hybrid FEA/LP model

Clearly the model produced results which closely match the experimental data in general form. However the model predicts that the solenoid actuates at 5V, but the experimental solenoid did not actuate until 6V.

The total actuation delay ($t_{dA}+t_{tA}$) from the model is compared below to experiment:

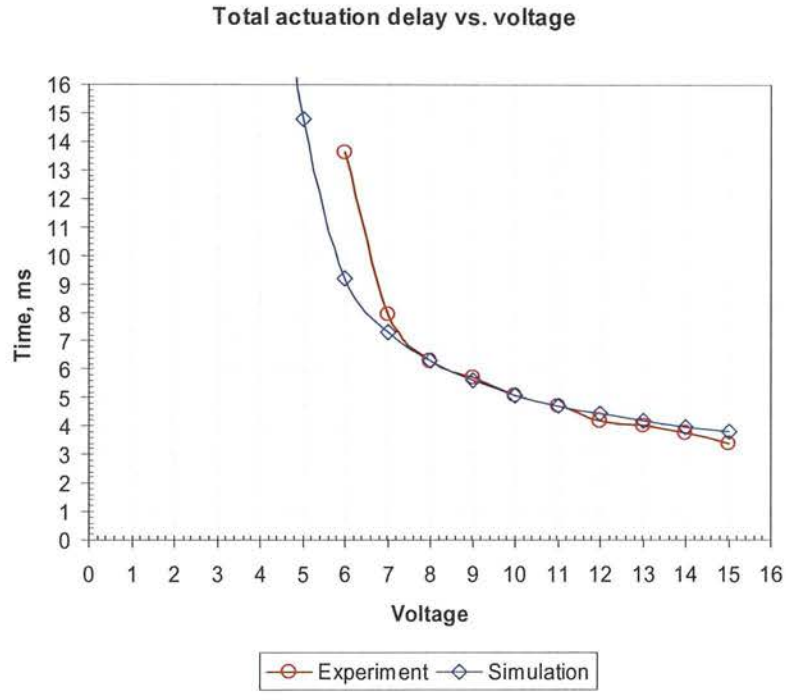


Figure 3-60: HPV total actuation delay

The most significant difference between the model and the experiments is seen at low voltages, with the model predicting substantially lower actuation delay than was measured. This could be due to measurement error of the spring pre-load, to which the results are very sensitive at low voltage. Overall the model shows the same trend as the experiment; between 7V and 14V the model is accurate to within 10%.

3.5.6 Model results for a typical motoring cycle

Once it was compared to the experimental results, the model was used to investigate the behaviour of the HPV when subject to the motoring cycle excitation sequence. It is assumed that the motor is rotating at 1800rpm, giving a duration of the motoring stroke of 16.7ms. The excitation voltage is 13.4V, and the FET gate is fed with 5ms pulse at 13.4V, followed by 13ms of pulsed operation at 35% duty cycle.

It should be noted that this is a rather artificial simulation because there is no fluid damping modelled nor forces from the fluid flowing through the poppet; there was no experimental data to allow these forces to be measured. The damping effect of flooding with oil was seen to

be small on the bench (<10% effect on total actuation delay), but in operating DDPMs there is clear indication that fluid forces are significant.

The figures below show the results.

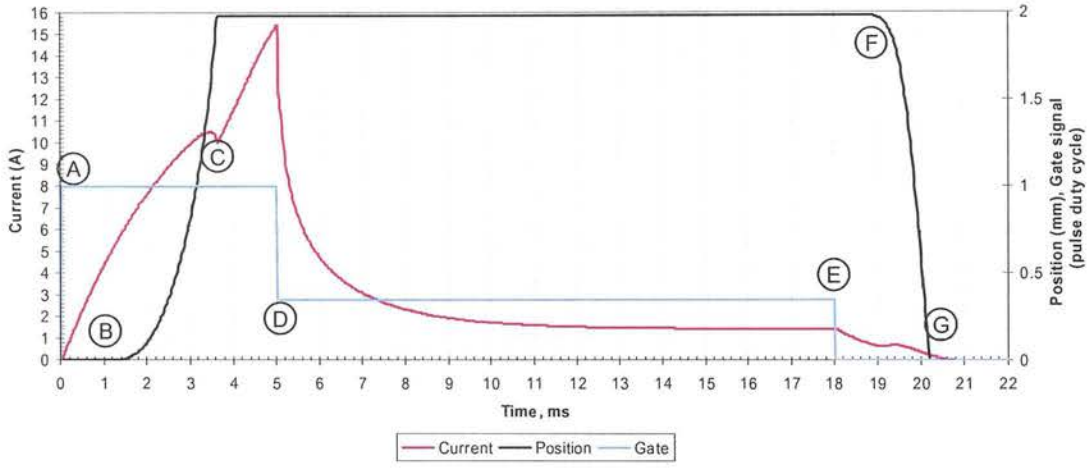


Figure 3-61: Simulation of HPV motion during DDPM cycle at 1800rpm; position.

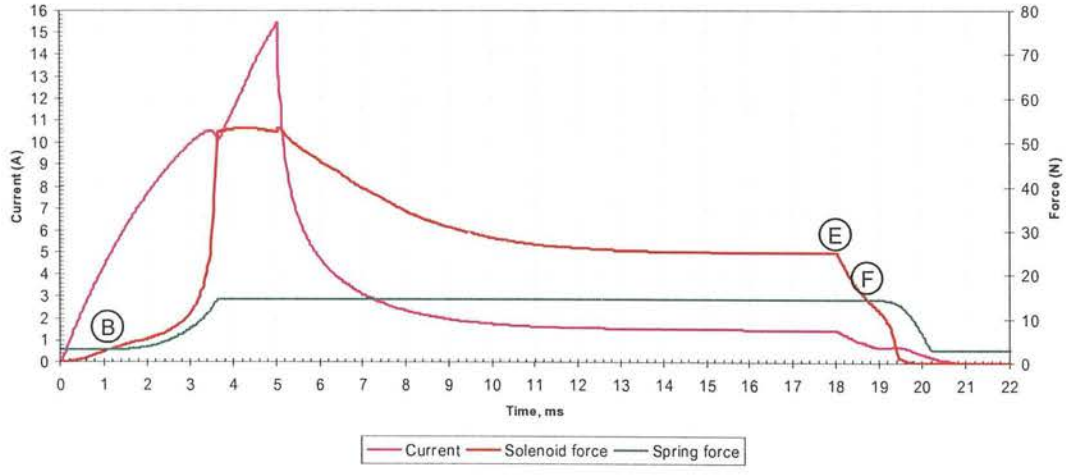


Figure 3-62: : Simulation of HPV motion during DDPM cycle at 1800rpm; force

Referring to the labels in the figures above, the following events occur in the model:

(A) the FET switches on and the current starts to rise

(B) When the solenoid force exceeds the spring preload force, the poppet starts to move

(C) The armature hits the pole causing an inflection in the current trace

(D) The FET gate is switched to pulses at 35% duty cycle; the current starts to decay to a steady-state latching current.

(E) The FET pulses cease and the current starts to decay sharply

(F) The solenoid force is less than the return spring force and the poppet starts to move back to the default position

(G) The poppet has returned to the default position.

3.5.7 Investigation of spring rate and preload

The model was used to investigate the effect of the return spring preload and rate on the total actuation and release time delays of the HPV, with the same excitation voltage and FET gate signal as above. Three spring rates were investigated: 3, 6 and 9 N/mm, each with a range of preloads from -1mm to +1mm. The force from these springs is plotted below with the solenoid force (from FEA) at 500, 1000 and 1500 AT.

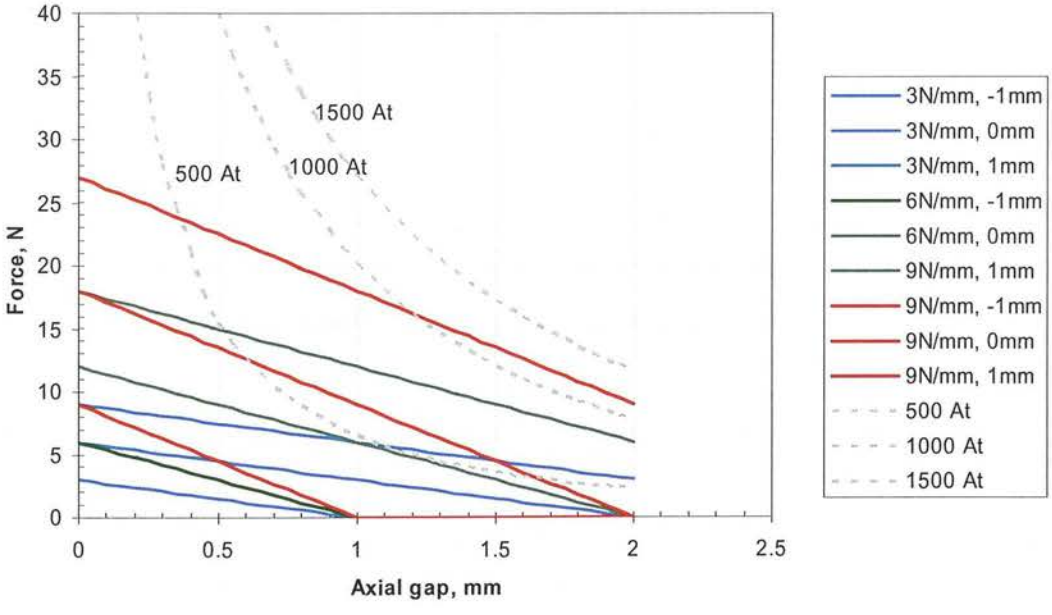


Figure 3-63: Spring and solenoid force

Clearly the force available from the solenoid is at a minimum when the axial gap is at a maximum. To obtain minimum release time, the maximum spring force is desired when the axial gap is zero. To obtain minimum actuation time, the minimum spring force is desired when the axial gap is at a maximum. A higher-rate spring is preferable to obtain both objectives.

The model was used to calculate the total release and actuation time for this range of springs as shown below.

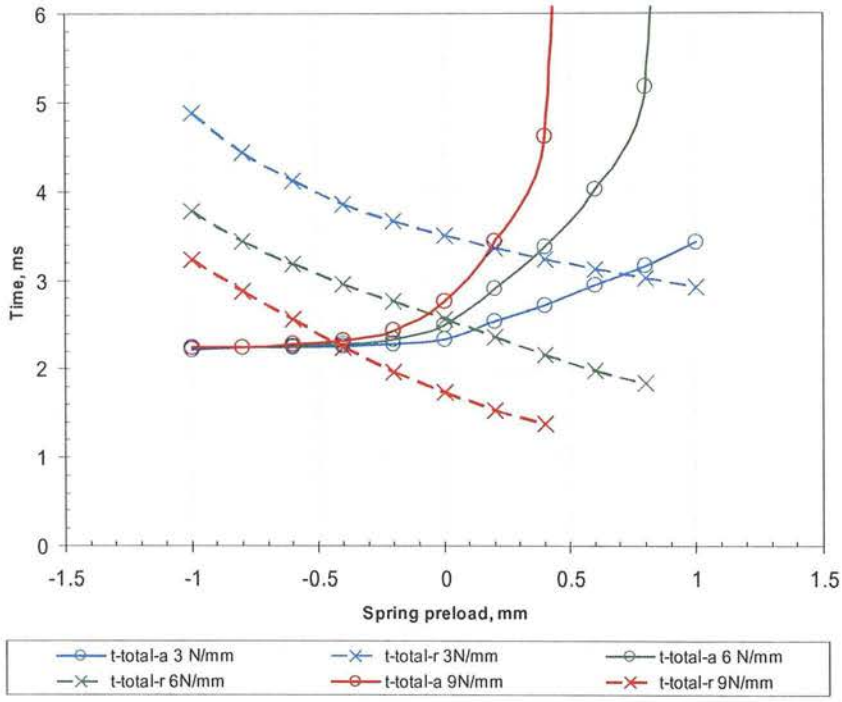


Figure 3-64: Model results of spring rate and preload on total actuation (t-total-a) and release (t-total-r) delays

At all preloads, as spring rate increases, the actuation time increases and the release time decreases. With a negative preload, the actuation time is not strongly affected by spring rate; by the time the armature has moved to the point where the spring force starts, the solenoid force is much increased.

The ideal spring rate and pre-load depends on balancing the objectives of minimum release and actuation delays. It has been shown in section 3.2.6 that it is the release time that is most important. By this measure, a higher rate spring with a zero or negative preload is desirable. However a negative preload is unacceptable because this would leave the default state of the HPV undefined, and the presence of fluid damping requires that a return force always be present. The choice made for the DDPM HPV of a 5.8N/mm spring with a 0.5mm preload seems reasonable in hindsight by these criteria.

3.5.8 Conclusion of the transient analysis

A hybrid FEA/LP model has been constructed for the HPV solenoid. This shows reasonable agreement with the limited amount of experimental data available and certainly displays all of the qualitative behaviour exhibited by the real solenoid on the bench, although fluid forces in an operating DDPM are not considered. The model has been used to draw some general conclusions about the return spring rate and preload.

3.6 Conclusions

The objectives set out at the start of this chapter have been achieved as follows:

- A review of valve functionality requirements for the DDPM operating cycle has concluded that the most appropriate choice for a propel DDPM is a normally-open, solenoid-closed LPV, and a normally-closed, solenoid-opened HPV.
- Experimental results of a 6-cylinder DDPM running at constant speed, have shown that the reduction of transit times of the solenoid valves is of the utmost importance for accurate commutation.
- Purely analytical techniques are limited for application to the DDPM solenoids due to non-linear magnetic effects. Parametric magnetostatic FEA has been demonstrated to be a powerful technique to optimise the electromagnetic performance. Using this method, a four-fold increase in magnetic force was achieved for the HPV while retaining the same outside dimensions. Experimental results with the prototype valve confirmed the predictions of FEA, and led to a significant improvement in overall DDPM performance.
- Valve prototypes have been described for the propel DDPM with the objectives of minimising moving mass, and maximising solenoid and spring forces. The critical mechanical design features and manufacture of these valves has been discussed. Experimental results have been presented.
- A hybrid FEA/lumped-parameter dynamic model has been described and validated with experimental results for the improved HPV, subject to transient voltage excitation. The model has been used to draw conclusions about the choice of return spring rate and pre-load for best performance.

4 The DDPM propel demonstrator

4.1 Introduction

The hydrostatic transmission is well-established as a means of transmitting the power of an internal combustion engine to the wheels of off-road vehicles such as wheeled loaders and forklift trucks. As leading manufacturers of such systems, Sauer-Danfoss were interested in investigating the potential of the DDPM as a vehicle propel motor. To this end they supported the creation by AIP of the first DDPM suitable for vehicle propel, and a vehicle to demonstrate its capabilities. This chapter describes this work.

This aims of this chapter are to:

- describe the fundamental characteristics of the DDPM as applied to vehicle propel and the implications for the design of secondary-controlled hydrostatic transmissions;
- describe the development of the mechanics and the embedded control software;
- describe a demonstrator vehicle propelled by a DDPM, supplied with fluid from a DDP connected to the engine;
- present experimental results from this vehicle.

The DDPM propel demonstrator vehicle described in this chapter was later converted into the DDP propel demonstrator described in Section 5, where some of the details of the mechanical installation are presented in more detail.

4.2 Secondary control of hydraulic motors

Conventionally, hydrostatic transmissions fall into two categories (Bauer, 2005) which between them account for the vast majority of installations.

In closed circuit systems, the flow from a displacement-controlled pump controls the speed of a hydraulic motor. The pressure in the system is linearly related to to the integral of the differences between the flow provided by the pump, and that consumed by the motor; typically such systems are hydraulically stiff. The roles of pump and motor reverse going downhill or during deceleration, delivering energy back to the engine. The fluid flows in a continuous loop between pump and motor. Direction reversal is usually achieved by an over-centre swashplate; a separate charge pump flushes this loop to provide cooling, filtration and

guarantee positive pressure on both hydraulic lines. Such systems are usually employed where the propel function is the main consumer of energy, such as in wheel loaders (Komatsu 2003).

In typical open circuit systems the motor is supplied with fluid via a proportional directional control valve. Often these valves are load-sensing, sending a demand pressure signal to the variable displacement pump. Upstream of the proportional valve, the pressure is determined by a pressure control loop implemented as a swashplate pressure compensator. The speed of the motor is controlled by the valve introducing a pressure drop between pump and motor, typically a spool valve in series with a compensator configured to maintain constant pressure across the spool. The same pump may also supply working functions of the machine. The motors may be fixed, or (less commonly) variable displacement but most case it is the control of the proportional valve that determines the vehicle speed. Such systems are usually employed where propel is secondary to the working functions, such as aerial work platforms.

A third category i.e. 'secondary control' has been investigated by researchers but so far found only limited application (Bauer 2005).

In secondary-controlled vehicle propel systems the motion of the vehicle is controlled by manipulating motor displacement. At the simplest level this can take the form of a single pump/motor connected to the driveline of a vehicle, and capable of pumping into or motoring from an accumulator. Here the pressure is fixed by the state-of-charge of the accumulator, and motor displacement is varied (taking into account the pressure) to control the amount of torque added to or extracted from the vehicle driveline; typically there is no speed control loop. Hydraulic "parallel hybrid" systems of this configuration offer the possibility to reduce fuel consumption of heavy highway vehicles such as delivery trucks, without the weight and expense of electric systems (Wu et al. 2004).

To create a complete secondary-controlled hydrostatic transmission, a pressure-controlled pump supplies fluid to one or more motors connected on a common high-pressure line. This provides the following features:

- multiple wheel motors can be supplied from a single pump yet still be individually controlled, allowing 4-wheel traction control (Liebenberg 1991);
- a distributed "ring-main" system can be created where energy recovered from one motor can be supplied to another (Nikolaus 1981);
- hydraulic motors which are located a long way from the prime mover can be controlled without suffering from oscillations caused by the large compliance of the connecting hose (Bauer 2005).

Secondary controlled systems are often designed to have very low hydraulic stiffness, an accumulator being added between pump and motor with the aim of maintaining a constant pressure (Nikolaus, 1981).

In a primary-controlled transmission, control of vehicle speed is achieved by varying the pump flow. In a secondary-controlled transmission, vehicle speed control requires a feedback control loop which acts to vary the displacement of the motor. Speed control of a vehicle with a secondary-controlled axial-piston motor has been demonstrated by Kim and Lee (1996).

4.3 DDPM vehicle transmission concepts

It was intended at the start of the propel DDPM development that the machine would be demonstrated in a secondary-controlled hydrostatic vehicle transmission. The system concept is shown below.

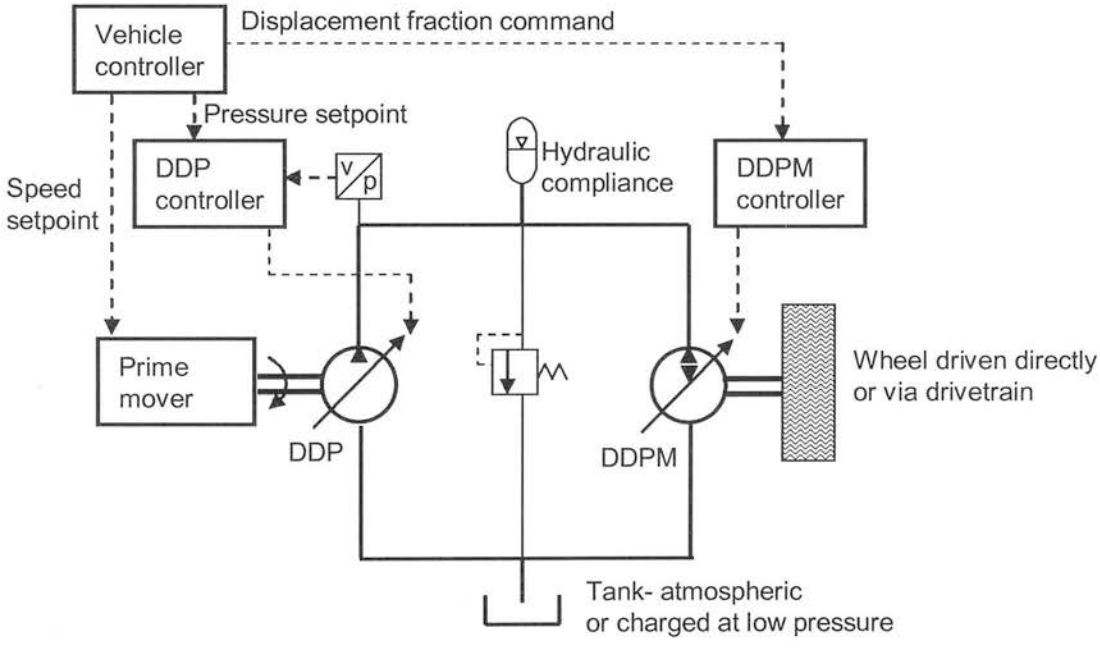


Figure 4-1: DD secondary-controlled hydrostatic transmission concept

The DDPM drives the wheels either directly or via a geared drive-train. The DDP supplies fluid into the high-pressure line, which has a controlled amount of compliance to smooth the pulses from the DDP. The DDP works in pressure control mode using a feedback signal from a pressure transducer. The vehicle system controller receives the driver demand and uses this to calculate a torque demand at the wheels. The displacement of the DDPM is set such as to satisfy this torque command at the present pressure level in the high-pressure line. As the torque is the product of displacement and pressure, the vehicle controller has the choice of

choosing the DDPM displacement and the DDP pressure which together produce the torque demanded. This choice is made on the basis of the following constraints:

- the maximum pressure is set by the maximum pressure rating of the system;
- the minimum pressure is set by maximum displacement of the DDPM and the flow limit of the DDP at the current engine speed.

Between these constraints the controller can choose what is optimal regarding efficiency or some other criterion.

During braking, the DDPM pumps fluid into the high-pressure line, charging the accumulator, pressure being limited by the system relief valve (not shown on the schematic above). The energy stored during braking is stored in the accumulator for later re-use.

4.3.1 'False rev' motor starting concept

The DDPM operating cycle described by Salter and Rampen (1992) has a drawback when applied to vehicle propel. In order to open the HPV, energy must be extracted from the shaft to raise the cylinder pressure to that of the high pressure port, because the HPV solenoid is not strong enough to open against any significant pressure. If the vehicle is stationary, the shaft is stationary, and this method cannot be used to open the HPV. Therefore the DDPM as originally conceived is not capable of starting from zero speed.

Rampen invented a scheme to address this issue. A 3 port, 2 position switching valve is inserted into the high pressure supply to the motor. When energised, this connects the high-pressure port of the motor to tank while blocking the connection to the high pressure supply. As this valve suddenly dumps the pressure in the volume within the DDPM, it is called the 'dump valve'. With the high pressure port of the DDPM now de-pressurised, the valves of the DDPM can be set in the correct configurations depending on the angular position of the eccentric, such as to produce torque in the desired direction of rotation. Once the DDPM valves have been set up, the dump valve is de-energised and the motor produces torque. Once the vehicle starts moving, the original DDPM motoring cycle can be used and the dump valve is not required until the vehicle speed again reaches zero.

As the valve set-up sequence followed the same sequence as if the motor was turned one complete revolution, this method was named the 'false rev'.

4.4 Overall mechanical design

The propel DDPM was based on the components already developed for a 12cc/rev fixed-speed DDPM as shown below.

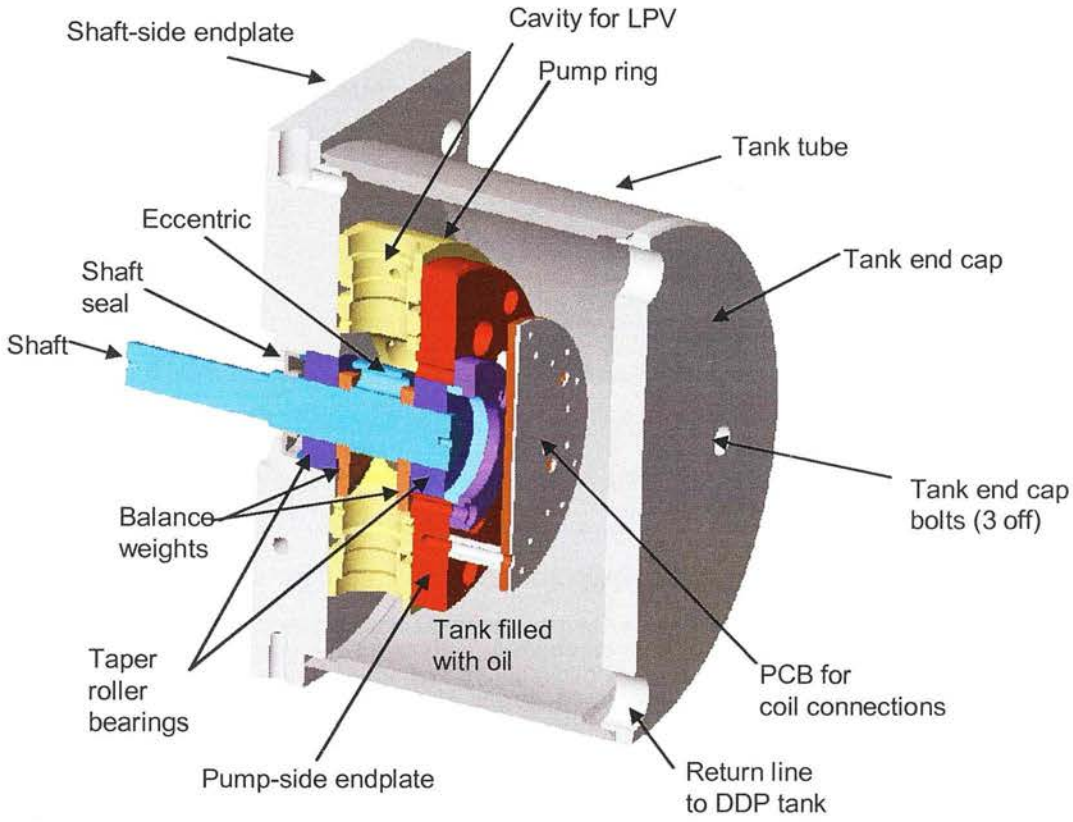


Figure 4-2: Section of the propel DDPM

Crankshaft bearing

The earlier fixed-speed DDPM had used Glacier DU plain bearing bushes to support the crankshaft. Operating at 1500rpm gave a surface velocity of 3m/s, and hydrodynamic lubrication was sufficient that very little wear was ever observed on these bearings. There were no external shaft loads because the machines were always coupled to an electric motor with a flexible coupling.

The propel DDPM had to operate down to zero speed, while resisting substantial external shaft loads from the belt-drive arrangement in the demonstrator vehicle, so taper roller bearings were selected.

Valves

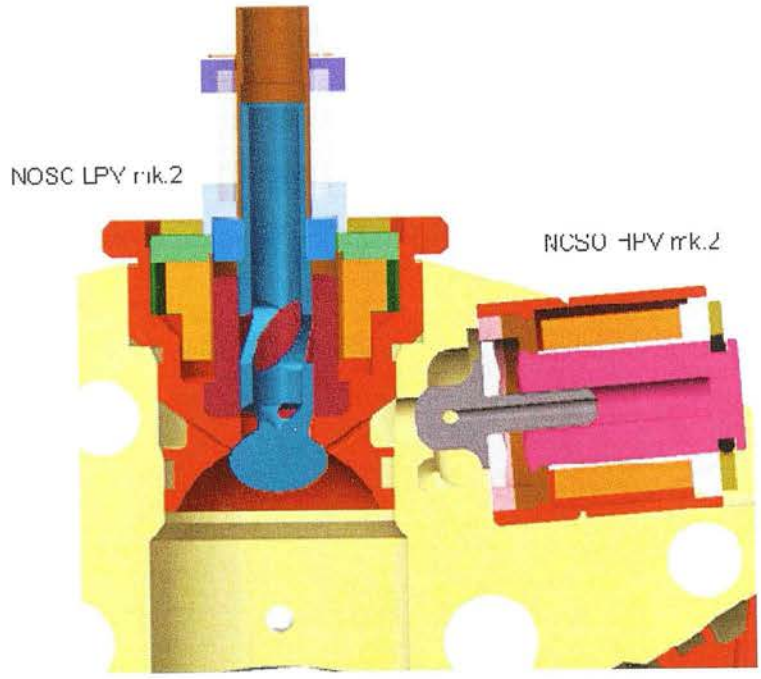


Figure 4-3: Valves for the propel DDPM

The details of the development of the commutating valves for the propel DDPM are given in Section 3. The HPV was fully developed during the high-speed DDPM work and was considered to be a "black box" for the propel work. Initial tests in the laboratory were made with the mk. 1 NCSO LPV, but this was upgraded to mk.2 before tests in the vehicle.

Pumping group material and process selection

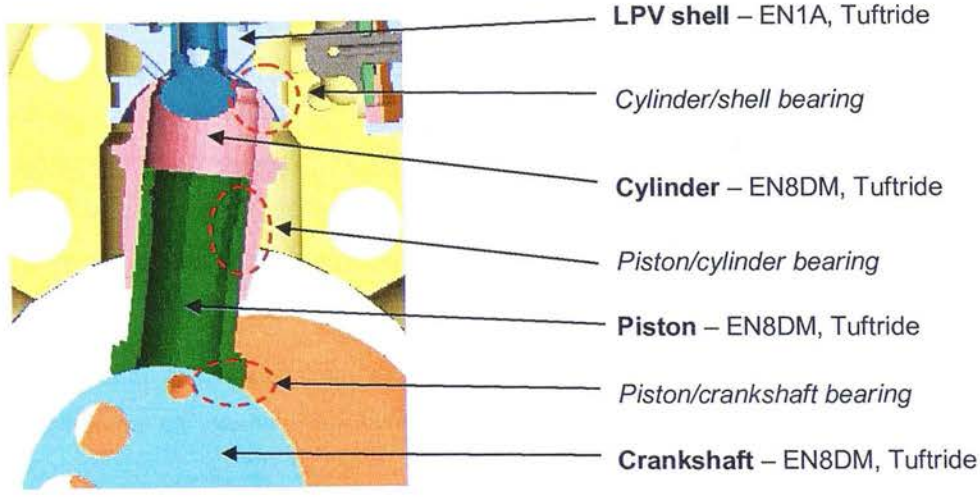


Figure 4-4: Features of the radial-piston mechanism used in the propel DDPM.

The kinematic arrangement of the spherical cylinder/shell bearing and the cylindrical piston/crankshaft bearing effectively removes side loads from the piston/cylinder bearing. This is unlike the swashplate pump design in which such loads are intrinsic. Although the piston/cylinder interface is not loaded, it still needs careful design because the fit of this interface is crucial to operation of the machine. A loose fit will cause high leakage, while a tight fit risks jamming under the influence of fluid contamination or differential expansion of piston and cylinder due to fluid pressure. This latter effect influenced the design of the tapering wall thickness on the cylinder.

As the machine would work down to zero speed, hydrodynamic lubrication could not be expected to separate the loaded surfaces with a fluid film at all times. It was therefore important that these loaded bearing surfaces be hard, to resist wear.

The crankshaft, pistons and cylinders were EN8DM medium-carbon steel. The LPV shell was 0.1% carbon steel (EN1a) for better magnetic properties. To remove the risk of distortion associated with case-hardening, all these parts were treated with "Tuftride" low-temperature salt-bath nitriding. This allowed the parts to be turned soft, then lapped before the Tuftride process – there was no need for finish grinding after hardening. Tests with a Leitz micro-hardness instrument showed a case hardness equivalent to a tensile strength of 1600N/mm2 (four times the base material hardness), reducing to the base material hardness at a depth of 0.50mm.

4.5 Hydrostatic bearings

On the fixed-speed DDPM, both the cylinder/shell bearing and the piston/cylinder bearing were over-clamped hydrostatic bearings. These were satisfactory at 1500rpm, where there was 3m/s of shear velocity to generate hydrodynamic lift. However early tests on the propel DDPM indicated that the piston slipper exhibited excessive stiction when starting from zero speed under load. It was found that it was very difficult to define a combination of parameters that achieved an acceptable level of both stiction and leakage. The design was changed to an underclamped pad stabilised with a capillary impedance, which reduced stiction by a factor of 6. The shear losses at high speed were unaffected.

4.5.1 Review of slipper pad literature

The overclamped hydrostatic bearing has been used on the piston slipper of axial-piston pumps for over 50 years, and there is a considerable body of literature relating to them (Hooke and Li 1988). As axial piston machines are mostly used as pumps, there is little literature on the breakaway friction characteristics.

In axial-piston pumps, the slipper pads are subject to off-axis piston loading because the machine usually operates with a non-zero swashplate angle. The kinematics of the radial piston arrangement used for the DDPM keeps the piston load normal to the pad at all times – this is usually referred to as "centrally loaded". The spherical bearings in the DDPM are very much further away from the slippers than is the case for axial-piston pumps, so tilting moments from spherical bearing friction is much less significant. Tilting moments can still occur due to the shear force acting tangentially on the piston slipper. Koc et al. (1992) state that for the centrally loaded slippers they investigated, the gap remained fairly constant all round the slipper i.e. there was little tilt observed.

Koc et al. (1992) also report that the observed steady-state behaviour of these bearings in use is extremely complex, being influenced by surface roughness, fluid viscosity and very subtle changes in pad profile. According to early theories the simple flat pad should not work at all, because it has no stability with regard to tilting moments. A consensus has emerged that the reasons they do work is down to very subtle ("0.1 to 10 microns") convexity of the pad. Koc et al. (1992) suggest that regardless of how flat they start out, slippers acquire the required non-flatness during the running-in period. Non-flatness may also be generated by the elastic strain of the piston pad, which at the scale of the oil film is very significant.

For a designer looking for ultimate efficiency it may be tempting to evacuate the case, eliminating churning losses. But another consensus (Koc et al.1992) seems to be that flooded conditions are absolutely necessary for slippers to work; the leakage flow from the pad is

insufficient to provide hydrodynamic lubrication. Case flooding also helps to conduct heat from the bearing, important at high speed and pressure, when local oil heating under the pad may cause smaller gaps than expected, and therefore a risk of asperities coming into contact.

In axial piston pumps the slippers are subject to dynamic loads due to inertial effects, friction in the spherical bearing and alternating cylinder pressure from commutation, amongst others. These effects have been studied in time-domain computer studies (Harris et al. 1996), which show that the alternating pressure load causes the pad to behave somewhat as a squeeze-film bearing, the gap opening up during intake and closing up under pressure. These studies also predict that the pads make edge contact at some points in the cycle. The simulation results seem to explain "witness marks" observed when a hard-worked axial piston pump was examined.

The design of a hydrostatic bearing involves a compromise between shear losses and leakage losses. Canbulut et al. (2004) investigated the leakage from slipper pads in axial piston pumps and the complexity of the behaviour they observed led them to train a neural network to predict the results.

The significance of leakage is greater for the DDPM than simply energy loss. The more the chambers leak, the more the timing has to be advanced to guarantee pressurisation. For the "false rev" propel DDPM, at a given pressure there would always be a minimum creep speed below which it was impossible to open the HPV, because the leakage from a closed chamber would exceed the volumetric contraction of the cylinder. This meant that the propel DDPM bearing design had to be biased towards low leakage rather than low shear forces.

For a large high-speed DDPM, Salter and Rampen (1993) proposed that the tilting moments in the direction of rotation be dealt with by machining two pads on the piston slipper, each fed separately via an impedance from the chamber. This provides full hydrostatic stability to the pad to tilting moments. This was investigated experimentally (Rampen et al. 1995). The scale of the pads used in the first propel DDPM seemed to preclude the application of this technique. Salter and Rampen (1993) also proposed a square pad for better rotational stability and this idea was adopted in the DDPM development. However their proposal for an asymmetric pad shape, designed to increase hydrodynamic lift in the unloaded condition, was not applied because the main focus was on starting torque, where hydrodynamic effects do not apply. Recent tests by Uwe Stein of AIP indicate that when off-loaded, symmetrical square piston pads (similar to the SD1B machine) tend to operate with the toe (leading edge) digging into the eccentric, preventing the formation of an effective hydrodynamic wedge. This explains the apparently frictional behaviour observed in the idle losses tests of the SD1B

(Section 2.5.3). The asymmetric bearing proposed by Salter seems to offer the lowest possible losses but may only be practical at a larger scale than the machines described here.

At the start of the propel DDPM development, it was intended that a circular over-clamped pad be used, as in the fixed-speed DDPM described in section 3.

4.5.2 Overclamped bearing principles

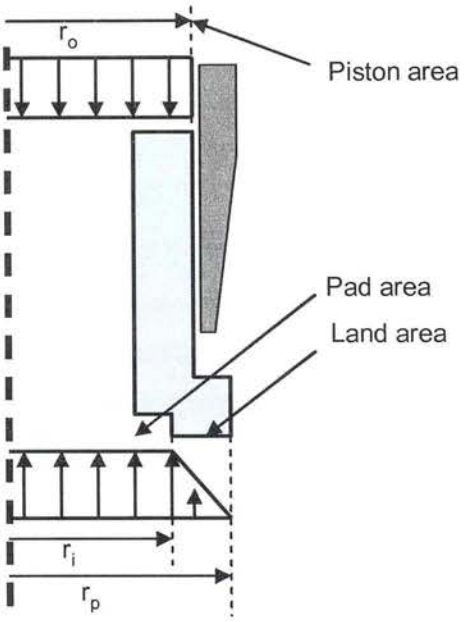


Figure 4-5: Basic principle of a circular overclamped bearing

The principle of the overclamped hydrostatic bearing is shown above for a simple circular pad on a flat surface. The piston area is exposed to the full cylinder pressure, while the outside is at case pressure (presumed here to be atmospheric). The land is assumed to be separated from the running surface by a thin oil gap, through which leakage flows in a diverging viscous flow field, causing pressure under the land to decline between the pad and the outside. The "clamping ratio" Cr is the ratio of force from the piston area, to the force from the pad area plus the pressure distribution over the land area. The pad is "overclamped" when $Cr > 1$; this means that there is a residual hydrostatic force which acts to clamp the pad onto the running surface. This force is assumed to be balanced by the hydrodynamic effect on the land area, due to the sliding velocity. Koc et al. (1992) report that typical clamping ratios for axial piston pumps are 1.02 to 1.10.

It is assumed that the piston/cylinder gap is small enough to be neglected (typical clearance for the DDPM parts was 15 microns on diameter). For the given assumptions, there is a standard expression for the clamping ratio for an axial piston pump slipper, which accounts for the pressure distribution function (Koc and Hooke 1996):

$$C_r = \frac{2 \cdot r_p^{\,2} \cdot \ln(r_o / r_i)}{(r_o^{\,2} - r_i^{\,2}) \cdot \cos\alpha} \tag{53}$$

where α is the angle of the swashplate.

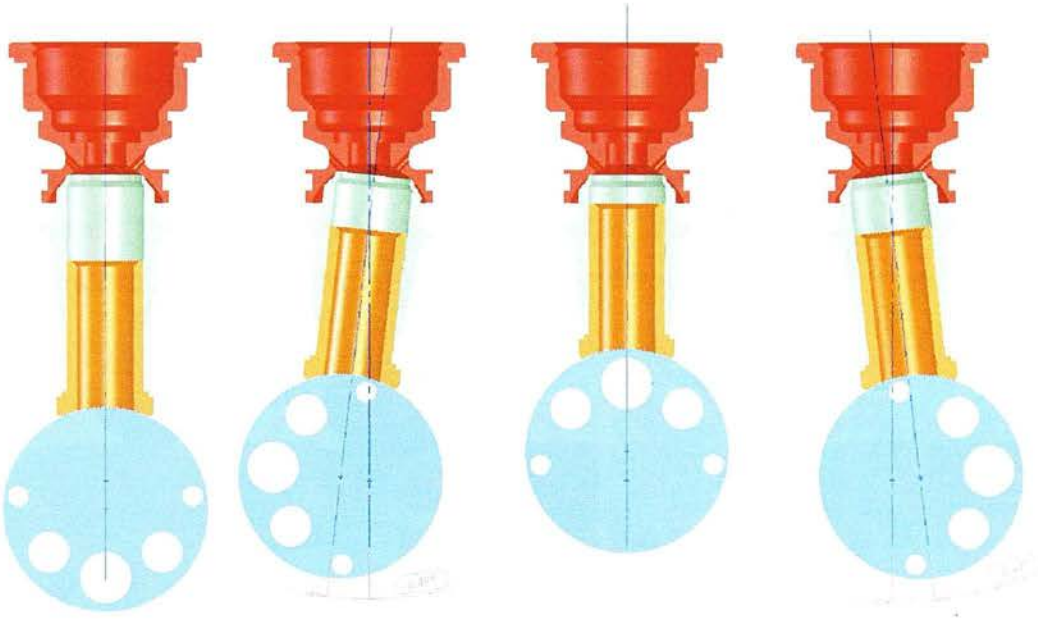


Figure 4-6: Articulation of the DDPM piston

The kinematic arrangement used on the DDPM shown above ensures that the piston load is always normal to the pad, so α=0 and the cos α term can be ignored. The slipper is always "centrally loaded".

The important dimensions of the fixed-speed DDPM bearings are shown below:

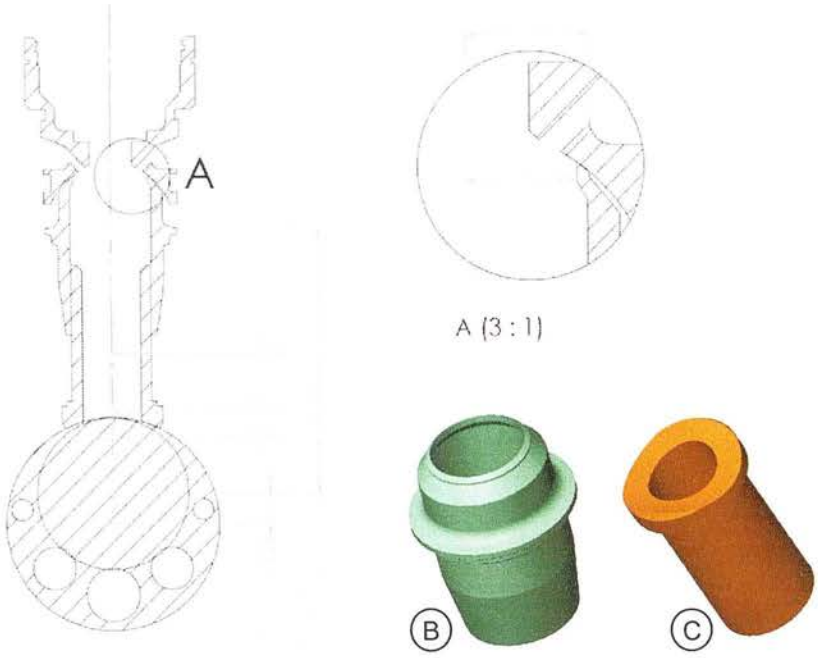


Figure 4-7: The over-clamped hydrostatic bearings used in the fixed-speed DDPM; (A) detail of cylinder bearing; (B) cylinder solid model; (C) piston solid model

For these dimensions, the equation above yields the following:

Cylinder/shell : Cr=1.035

Piston/eccentric : Cr=1.025

The cylinder/shell bearing was designed by Rampen with a higher overclamp for the following reasons:

- The features were so small that machining tolerances or wear could easily cause the overclamp to vary in service. If the overclamp fell to below unity, the cylinder would be unable to hold pressure. A higher overclamp provides a safety margin.
- This bearing operates at a much lower sliding velocity than the piston pad. Hence from an energy loss point of view, it is better to be designed to minimise leakage. A higher overclamp should reduce the leakage.

The piston pad overclamp of 2.5% means that the average pressure across the land of the piston is 41% of the chamber pressure. This is not 50% because of the divergent flow field in the significantly long radial thickness of the land – as shown by the logarithmic form of eq. 53. As the land gets thinner, average pressure tends towards 50%.

4.5.3 Early experience with circular pads

During early trials, the propel DDPM was working against a load comprising the inertia of a wheel and a gear pump working against an orifice. Under these conditions, the circular pads gave no trouble and attention was focussed on developing the control software. However as soon as the motor was tried in a vehicle, problems were observed. When starting on a hill, the motor often seemed to stick if there was insufficient pressure to accelerate immediately at start-up. The effect was consistent with a squeeze film bearing under the land becoming exhausted after a short time, whereupon the land touched down onto the eccentric and the gap disappeared. Once this occurred, the pressure distribution across the pad was undefined and the overclamp may have suddenly increased far beyond that predicted by steady-state theory.

This behaviour was tolerated while further controls development took priority, until a catastrophic failure occurred .

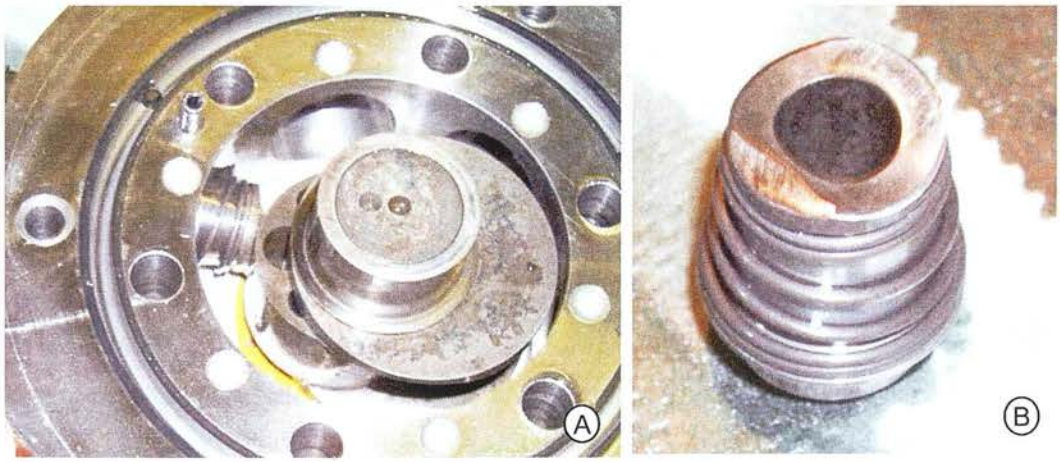


Figure 4-8: Catastrophic failure observed when starting on a hill; (A) piston and cylinder jammed between pump ring and eccentric; (B) impact damage.

It seemed that the tangential friction force on the pad was sufficient to cause the trailing edge of the land to dig into the eccentric. Once that happened, the piston jammed against the pump ring. It should be noted that at this time, the only mechanism keeping the pad in positive contact with the eccentric, apart from the clamping force of the bearing itself, was the spring which provided 20N at TDC. In later designs, AIP has used piston retention rings which guarantee the pistons stay in contact with the eccentric even in very cold start conditions, when the piston/cylinder viscous shear can exceed the spring force. This is a similar mechanism to that used by SAI in their radial piston motors.

It was concluded that there was a need to change from a circular to a square pad. This would provide more rotational stability due to more effective "wrap" around the eccentric; there were signs that the piston had rotated before digging in. Also, it would ensure that if the

piston did have a tendency to dig into the eccentric, the contact would be over a line rather than at a point, so more force could be resisted by the pad edge before plastic deformation of the piston or the eccentric.

The second conclusion was to investigate the stiction phenomenon in a special rig shown below.

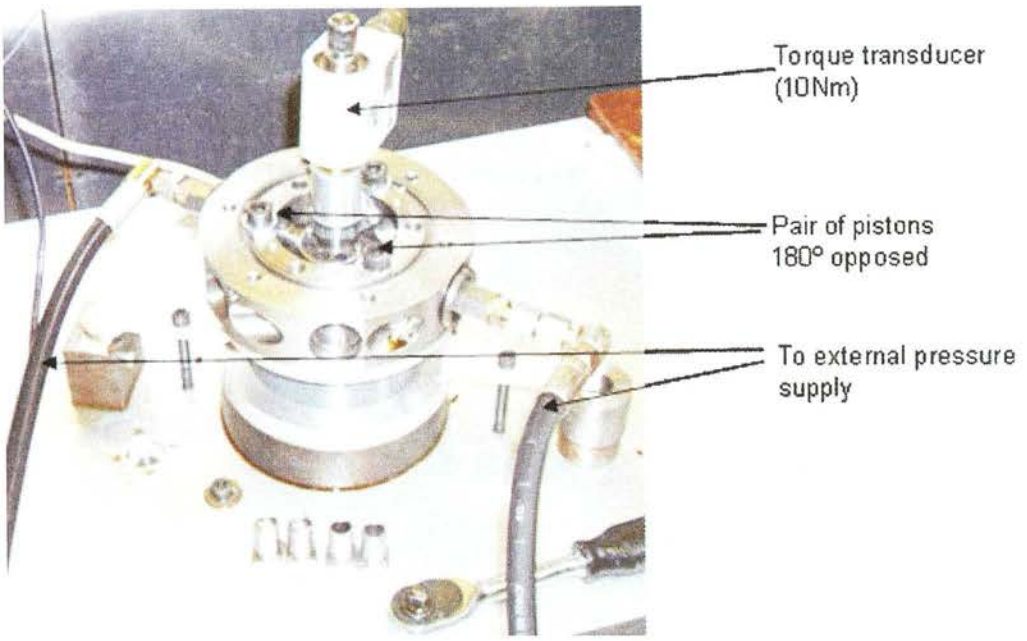


Figure 4-9: Stiction testing rig

Two pistons of the DDPM design were arranged in opposition and the eccentric was aligned with TDC of one of the pistons. As the area of both cylinder bores was the same, there was no net radial force on the shaft. However, whichever piston was aligned with TDC had an 80% higher lever arm than the piston at BDC. This allowed quick comparative tests by simply modifying the TDC piston and keeping the BDC piston the same.

Both pistons were pressurised by an external supply. After waiting 10 seconds for the squeeze film effect to diminish and steady state to be reached, the torque on the crankshaft was slowly increased by hand until the stiction was defeated and the crankshaft broke away. The peak torque which was exerted was captured by the torque transducer peak-hold facility. Each measurement was repeated five times.

The pressure of 140 bar was chosen for all comparative tests because it kept the torque for all tests within the 10Nm range of the transducer. The stiction torque was observed to be reasonably linear with pressure.

Tests showed that a pair of circular pads, of the dimension described above, caused 4.4Nm of torque, with negligible pad leakage. With a full complement of 6 pistons, on average 3 should be pressurised at any time, so this would translate to 6.6Nm in the motor.

The torque expected of an ideal 12cc/rev machine at 140 bar is 26.7Nm. However the actual effective displacement could be expected to be 10% lower than this (=24Nm) because there needs to be a timing margin built in to the control algorithm to guarantee pressurisation and depressurisation (section 3.2.6) . Out of 24.0N, the motor was losing 6.6Nm to stiction, giving a starting "torque efficiency" of 72%, assuming the roller bearing to be perfect and the spherical bearing to be insignificant. This is a poor performance for a piston motor.

Translating this figure to a tangential force, each piston was exerting 110N of tangential friction force. The piston force was 2470N. With the nominal 2.5% overclamp, a net clamping force of 62N would be expected. The tangential force from this depends on the friction coefficient.

Chapple (1992) quotes Rabinowicz that the coefficient of friction that can be expected between hard steel components at very low speed (0.01mm/s) as between 0.05 (perfectly lubricated) and 0.7 (dry). The flooded conditions in the DDPM crankcase are closer to perfectly lubricated than dry, so we might expect a maximum friction coefficient of 0.2.

In that case, 110N of tangential force equates to 550N of normal force. This gives the actual overclamp in the stiction case as 22%, meaning that the average pressure across the land is 22% of chamber pressure, as opposed to 41% from the classical overclamp equation. It seemed that under stiction conditions, the gap closed up until the asperities of the surface touched and the gap tended to disappear. At this point, the deflection of the piston pad may have been significant enough to determine the pressure distribution across the land. As the land in this case is radially thick, a small change in the form of the pressure distribution has a large change on the effective overclamp. The large effective overclamp also explains the negligible leakage observed.

4.5.4 Square piston design

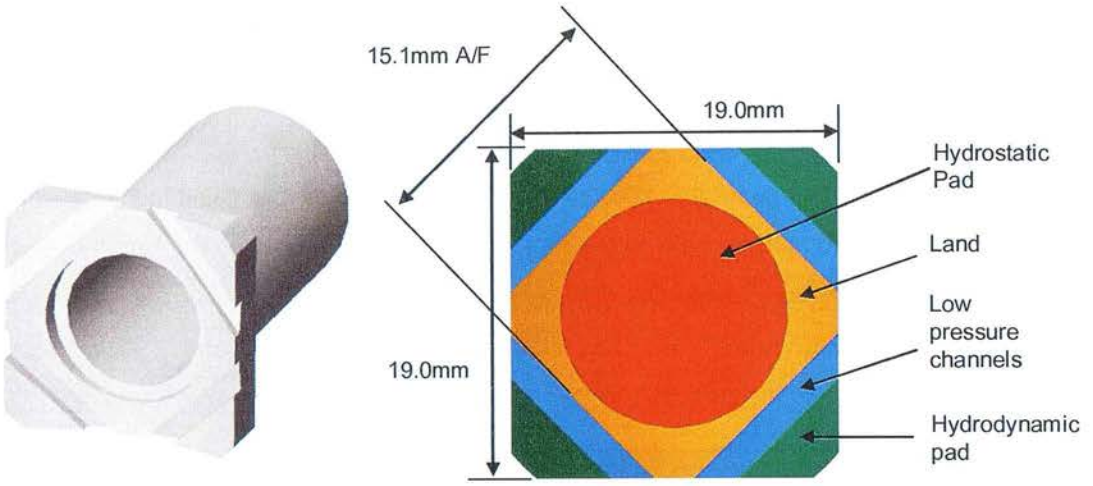


Figure 4-10: Concept of the square piston pad

It was obvious from the circular pad tests, that reducing the radial thickness of the land would reduce the increase of effective overclamp under stiction conditions. This observation has been noted by others (Manring et al. 2004). The square pad shape seems to offer the best resistance to rotation and digging into the crankshaft.

To try to satisfy these requirements, the author made a square pad with a "diamond" pattern of slots, machined with a 1/16" (1.58mm) end mill. The idea was that the corners of the pad would provide only hydrodynamic lift, the hydrostatic pressure being exhausted through the four slots.

To determine the ideal diameter of the pad, the stiction rig was used to test a pair of such pistons. The pad was progressively bored out in 0.1mm diametric steps, re-lapped each time and tested in the stiction rig:

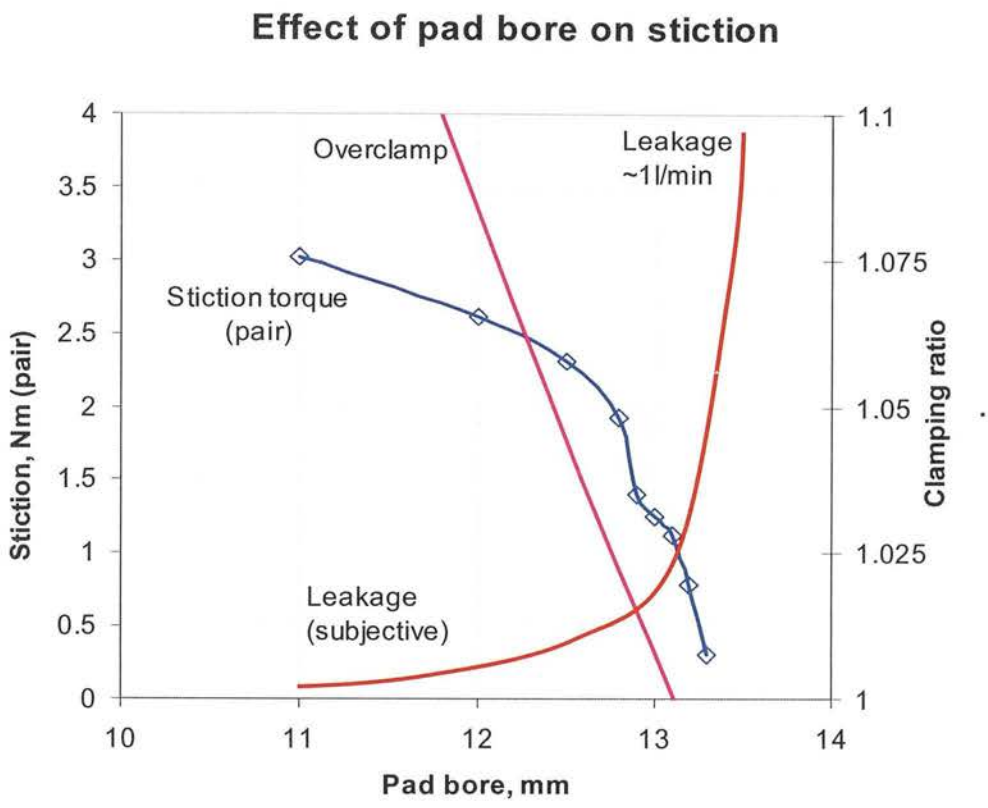


Figure 4-11: Stiction behaviour with the square pads; blue = stiction torque, pink = clamping ratio, red = qualitative leakage trend.

The theory on clamping ratio is described in terms of circular pads. This was applied to the diamond-shaped pad by measuring the projected area of the land plus the pad using the Solidworks geometry measurement tool, giving a total area of 226mm^2. This was equivalent in area to a circle with radius of 8.48mm, so this figure was used as r_o in eq. 53 (p.183). This will lead to some error in the overclamp calculations because the pressure distribution under the diamond land will not be quite the same as in the "equivalent" circular land.

Unfortunately the leakage from the bearing could not be isolated from the cylinder/shell leakage to be measured accurately, but the subjective trend is indicated above.

Between 11mm and 13mm, stiction declined slowly and leakage remained very low. With pad bores above 12.7mm (equivalent to a theoretical clamping ratio of 1.025), two trends were observed:

- The pad became more and more unstable. At 13.3mm, only the spring force of 20N kept the pad in contact, and the pad could be made to explosively lose pressure by applying a tangential force by hand.

- The leakage from the pad increased dramatically. At 13.3mm, the flow was about 1 l/min from each bearing.

In addition to the results presented above, there was a wide variation of stiction force depending on the surface quality of the piston and the eccentric. Pads with surfaces polished with 1200 grade paste gave up to 50% more friction than pads lapped with 400 grade paste. It seemed that a slightly rough surface helped retain the lubricating film.

Also observed was a very strong effect of the size of the lapping mandrel used on the piston pad. Pistons lapped on mandrels that were undersized or oversized by only 0.02mm relative to the eccentric gave about 50% more friction than those sized correctly.

It seemed that it was not possible with this design to achieve a combination of low stiction, low leakage and stability with respect to variations such as surface finish and manufacturing tolerances. The radially thin land might offer acceptable performance in a conventionally commutated motor, where the leakage can be tolerated, but in the propel DDPM the leakage was not acceptable. A target was set of one-tenth of the cylinder volume per second at 140 bar, which would allow the motor to operate down to a creeping speed of 6 rpm. This equated to 0.012l/min total leakage from the chamber- much less than observed with the square pad. It was concluded that the overclamped piston pad is fundamentally unsuited for use in a DDPM for propel duty.

4.5.5 Underclamped pad with capillary impedance

Following the above, Rampen suggested exploring the possibility of an impeded, underclamped bearing. Most of the literature on slipper pads relates to pumps, rather than motors, where the speed is usually sufficient to generate a significant hydrodynamic effect, and under those conditions, there appears to be no decisive advantage to either type (Koc and Hooke 1997). There is little mention however of the decisive advantage that an underclamped pad with impedance should have over an overclamped pad in the stiction case; the underclamped pad should stabilise towards a finite steady-state gap when loaded at zero speed, therefore the touching down behaviour of the underclamped pad should be avoided. Perhaps the reason as to why this is not emphasised more in the literature is that it is only valid for centrally-loaded pads – the underclamped bearing with impedance has little stability to the off-axis loads intrinsic in axial-piston motors.

Most of the literature on underclamped pads refers to them being stabilised by "orifices" of typically 0.4mm. The author could not imagine an orifice of that size achieving a significant pressure drop without a very significant and unacceptable flow rate. Considering the case of 140 bar chamber pressure, if the underclamp was 10%, that would mean that the pad was

exposed to 90% of the chamber pressure, and there was 14 bar across the orifice. Simple application of Bernoulli theory shows a flow velocity under these circumstances of 55m/s, giving 0.42 l/min with a 0.4mm orifice. This was 35 times as much as the target of 0.012l/min. In fact to achieve the target leakage, the orifice would have to be 0.068mm in diameter. Such an orifice, even if it was possible to make, would surely block very quickly.

The orifice is not ideal as an impedance because the pressure drop across it is proportional to density, not viscosity. To maintain a constant gap as the oil temperature changes, the impedance should be a capillary, the impedance of which decreases as viscosity decreases.

In fact the orifice described by Koc et al. (1992) was 0.4mm diameter, and 12mm long. With 30cst oil, the application of standard pipe flow equations show this generates a 10 bar pressure drop with 2cc/s flow, or 0.120 l/min, still 10 times the target. The Reynolds number is 200, and the "orifice" is actually working predominantly as a capillary in this region. Still, the simple drilled hole impedance seemed to be incapable of generating the required impedance level with practical dimensions.

At this point, Rampen was advised by Jonathan Gamble of Sun Hydraulics Ltd. of a method to create a capillary impedance that was resistant to blockage. Into a long drilled hole is inserted a wire, with a diameter slightly smaller than the hole. The wire is free to move around the hole, and the gap created is nominally an annulus but in practise will probably be a crescent. The calculations used for designing the capillary are given in Appendix 9.5.

A capillary was made comprising a 1.30mm hole, 20mm long, with a wire of 1.19mm diameter. The wire ends were bent over such as to allow the wire free movement within the hole – thought to be essential for resistance to blockage. With 14 bar across it, and nominally 32cst oil at room temperature, the flow was 0.2cc/s, measured by weighing the mass of the leakage oil in 100s. This indicated that the wire-in-hole impedance could be easily made to achieve the required level of impedance.

To prototype this, a square piston pad was used, but without the slots described above. The slots were abandoned on the basis that with an impeded bearing, the pad should never touch down, so the main reason to shorten the land, to have better control of the pressure distribution, was no longer important. With leakage being the dominant concern, the longer land of the full square pad should allow the pad to stabilise with a larger gap than would be the case with the thin land. This larger gap would help avoid asperities touching and therefore encourage purely hydrostatic separation. The pad was bored to 12mm, which equivalent circular pad theory predicted should have a clamping ratio of 0.81. The impedance was designed as a cartridge, retained in the piston by a circlip, and the gap around the outside sealed with adhesive.

The impeded pad was tested on the friction rig, with a friction of 0.52Nm for a pair at 140 bar. Neglecting other contributions, this equates to a torque efficiency, with three pistons pressurised, of 96.7%. The leakage was observed to be extremely low, but direct measurements could not be made. However a combination of the impedance measured and the predicted clamping ratio suggests that the leakage at 140 bar should be 0.2cc/s.

The design settled upon was a combination of 19mm square pad, bored to 12mm, with a capillary impedance comprising a 1.19 wire in a 1.30mm hole, 20mm long.

The figure below shows a section of the final impeded piston design, along with the other designs investigated:

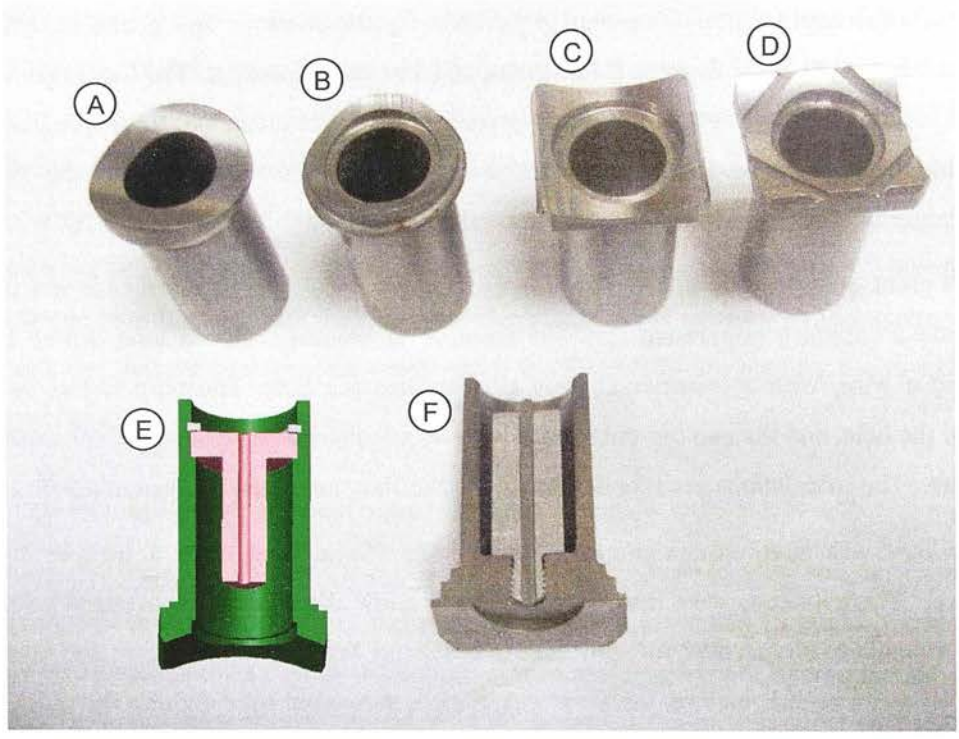


Figure 4-12: Piston pad development. (A) original circular pad; (B) reduced land; (C) square pad; (D) square pad with reduced land; (E) solid model of first underclamped square pad as used in the propel DDPM demonstrator; (F) section of a later design with a screw-in impedance cartridge

The underclamped pistons were deployed on the propel DDPM demonstrator. The starting performance was improved very significantly and stiction behaviour when starting was not observed after the pistons were replaced.

Subsequently, Dr Uwe Stein and Jack Lavender of AIP investigated the torque losses from the piston pads when pressurised at high speed. Their conclusion was that there was no

significant difference between the over-clamped and under-clamped designs, confirming experimental results from the academic literature.

4.5.6 Conclusions

- The leakage from the chambers of a propel DDPM must be extremely low to enable stable low-speed operation.
- The overclamped pad produces unacceptable stiction, unless the overclamp is marginal, in which case leakage is unacceptable and the bearing is unstable.
- The underclamped pad, supplied via a capillary impedance, causes only one eighth of the stiction compared of the overclamped pad. An annular capillary of practical dimensions can generate the required impedance. Leakage is within acceptable limits.
- The stiction benefit of the underclamped pad probably only applies where the kinematics of the machine ensure that the piston load is applied centrally.
- There is no significant difference in terms of viscous losses at high speed between underclamped and overclamped pads.

4.6 Electronic system

The overall electronic system design below was a collaboration between the author and Jon Almond of AIP, who did all the detailed circuit board design.

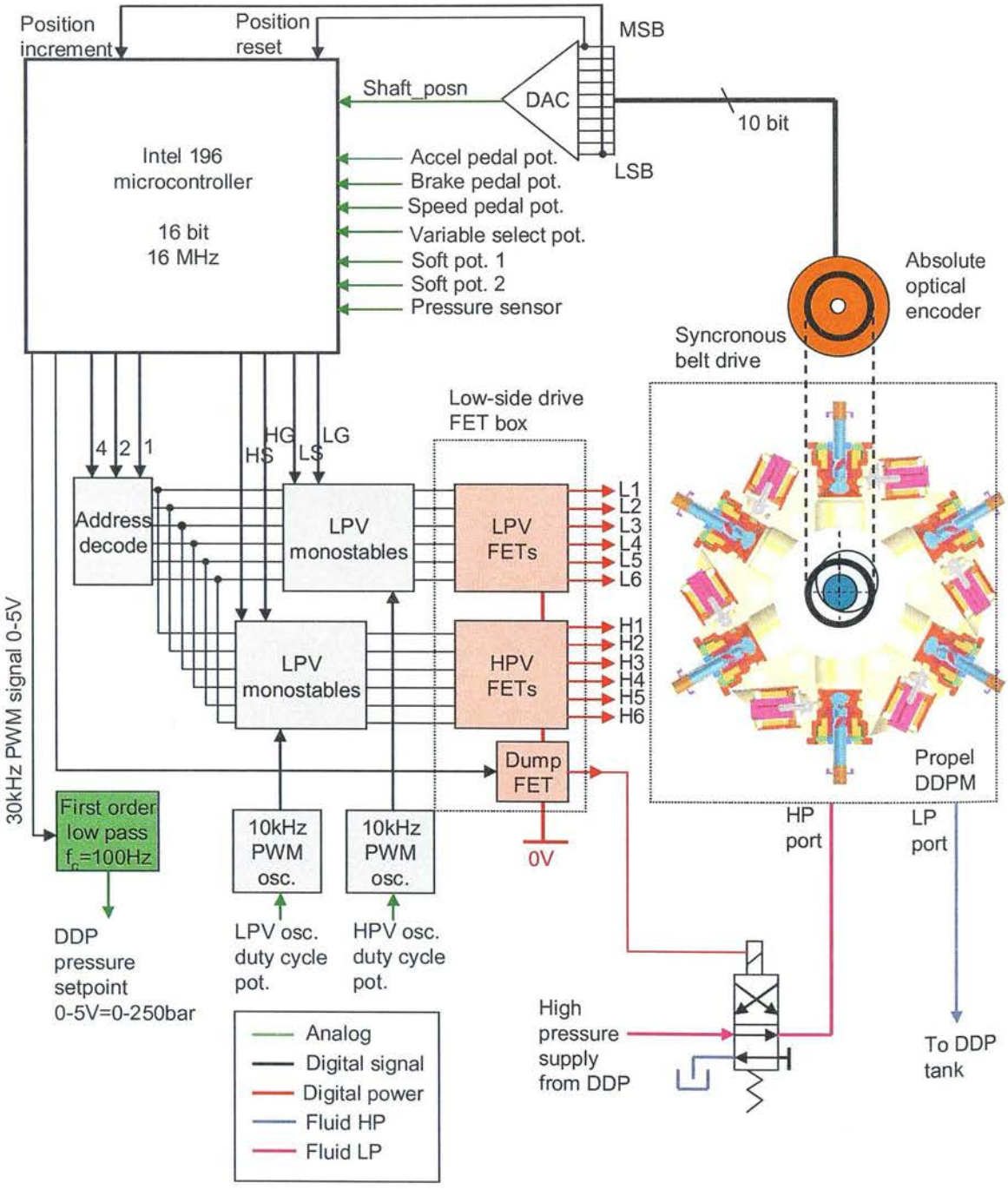


Figure 4-13: Overview of the electronic system for the propel DDPM

4.6.1 Controller

The Intel 80C196KB ("196") is a 16-bit microcontroller. The microprocessor at its core is from the 80x86 family, well-known for application in personal computers in the 1990's. Running at 16MHz, its calculation speed of 1-2 MIPS is very low compared to modern DSP-core controllers, it has no floating-point capability, and very limited on-board RAM. However the DDPM application makes little demand on the processor as there is very little "data" to process, and a modest amount of fixed-point calculations. Much more important are the hardware peripherals that are built into the chip:

- 8 analogue inputs, multiplexed to a 10 bit ADC;
- 4 counter/timers with interrupt capability, allowing accurate timing of periods, and interfacing to encoders;
- RS232 serial port;
- hardware pulse width modulation on digital outputs.

The chip was designed specifically for the automotive engine ECU application, which demands strict timing of fuel injector pulses relative to pulse position inputs from crankshaft and camshaft sensors. The DDPM control task is very similar.

4.6.2 Support electronics

The 196 does not have sufficient digital outputs to generate all the FET gate signals directly, so a multiplexing scheme was used. Monostable oscillators and a synchronous circuit controlled the initial pulse duration as shown below. This had the added benefit that the 196 only had to start and stop the pulse sequence sent to each FET gate, the synthesis of the waveform in between being done in external electronic hardware, helping to reduce software complexity and processor load.

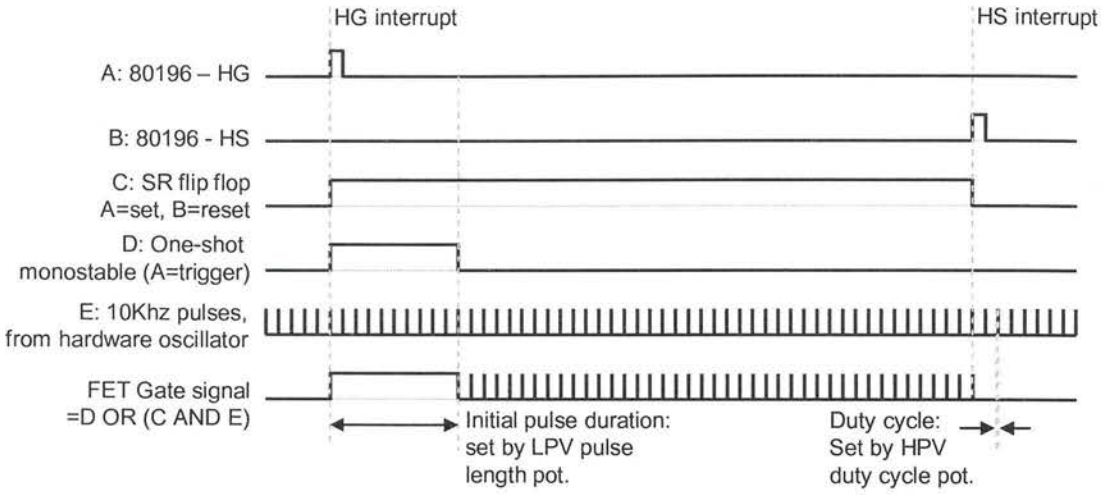


Figure 4-14: Generation of FET gate signals with dedicated electronic circuit

4.6.3 Power electronics

The same low-side driver FET arrangement was used as previously described for the fixed-speed DDPM (Section 3.2). The coils were wound for a 12V power supply, suitable for the golf buggy which was later used as a demonstrator vehicle for the DDPM. The flywheel resistor values were scaled appropriately.

One diagnostic tool turned out to be crucial for development. A small box on a trailing lead from the FET box, it contained 14 LEDs connected to the switched output of the FETs: 6 green for the LPV, 6 red for the HPV, one yellow for the dump valve and one spare which could be programmed to reflect the state of an internal software bit. The LEDs were arranged in a circle corresponding to the valve position in the DDPM. As the motor turned, the pattern of red and green lights "walked" around the display, showing the state of each valve. During the initial pulse of the sequence, the LED would glow brightly, becoming dimmer during the 10kHz pulsed phase.

Figure 4-15: Frames from a video showing the motor in the background, and the LED diagnostic tool in the foreground.

The power of this tool lay in the ability of the human eye to detect patterns, and deviations from them. Irregularity in starting or some other condition could be directly related to the signals being sent to the FETs. Control "glitches", such as the controller failing to switch off a valve it had switched on, could be easily diagnosed. The tool became less useful above 500rpm, when the rotating pattern in the LEDs could no longer be followed. Above this speed, a logic analyser was used for verifying valve control but was considerably less intuitive to use.

4.6.4 Shaft position feedback

The DDPMs made up to this time had rotated at a constant speed of 1500rpm. At this speed, and coupled to a prime mover with significant inertia, sufficient position measurement accuracy was achieved for commutation by means of a single pulse per rev of the shaft. This pulse reset the controller to a known angular position, and by measuring the period of rotation with a time-based counter the shaft speed could be calculated. Assuming the next rotation of the shaft took place at the same speed as the last rotation, the shaft position could be estimated between trigger pulses by means of a clocked timer. Interrupts were set on the basis of this timer to trigger the valve events.

The propel DDPM would have to operate down to zero speed, so a single pulse per rev was obviously inadequate. In addition, during the "false rev" at start-up the motor control had to know the angular position of the eccentric, so that the commutating valves could be set to the correct state. This meant that the controller needed an absolute angular position measurement.

This was achieved using a 10 bit absolute optical encoder, phase locked to the eccentric position by a synchronous drive belt. This position signal was converted to an analogue value by a digital to analog converter (DAC) because the controller did not have enough digital inputs to read every one of the 10 bits. When the motor was commanded to start, this analogue value was sampled to give a shaft position estimate. Once the motor started to rotate, the shaft position register was incremented by the least significant bit (LSB) of the encoder, giving 10 bits of shaft position accuracy. The valve events were triggered by interrupts based on this shaft position register, so were angle-based rather than time-based. Due to the analogue conversion stage, the starting position estimate was subject to a small error. To eliminate this error, the shaft position register was reset on the rising edge of the encoder most significant bit (MSB) on every subsequent revolution of the shaft.

Later on in the development, a weakness of this scheme was detected. If the vehicle was on a hill, then the motor did not always move in the direction demanded immediately after the DAC signal was sampled. The controller would interpret LSB pulses as movement in the desired direction, which may not be the actual direction if the vehicle was rolling backwards down a hill. This is one of the reasons for the motor starting problems detailed later on.

4.7 Motor software development

The propel DDPM software was built in a series of hierarchical layers. From the lowest level to the highest, the software comprised the following functions:

1. The starting sequence ("False Rev").
2. Encoder-synchronised valve events.
3. The cylinder-enabling displacement algorithm.
4. The torque control loop.
5. The speed control loop.
6. High-level vehicle functionality such as starting and direction reversal.

The source code of the propel DDPM embedded software is presented in Appendix 9.7.

4.7.1 First propel DDPM tests

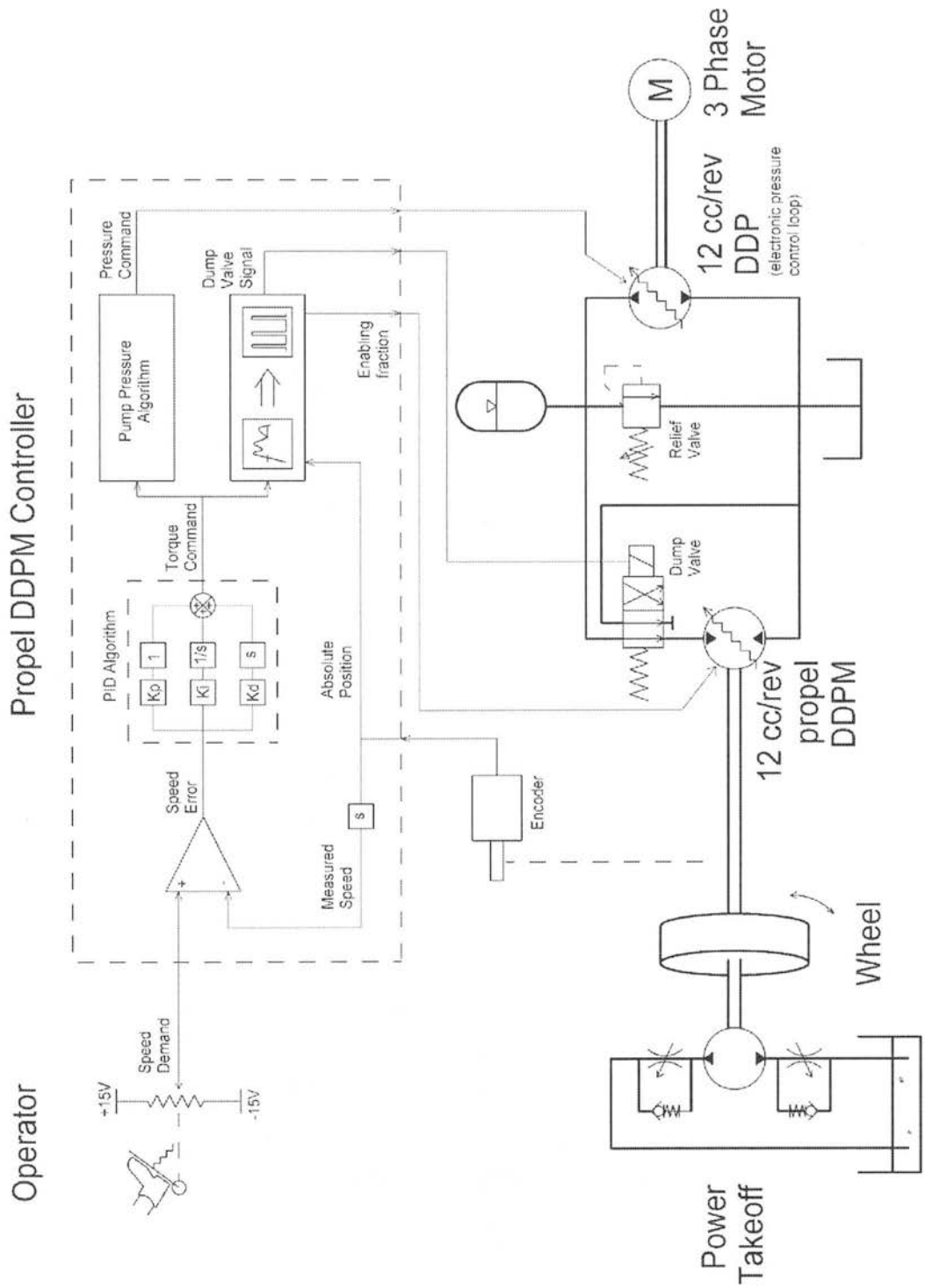


Figure 4-16: Schematic of the system used for commissioning propel DDPM, also showing the later speed controller

At the early stages of development, the DDPM was loaded by a gear pump working against an orifice, and an inertia in the form of a truck wheel and tyre. A DDP power-pack with a pressure-control loop supplied fluid at a constant pressure. Adjusting the needle valve stabilised the motor at a constant speed.

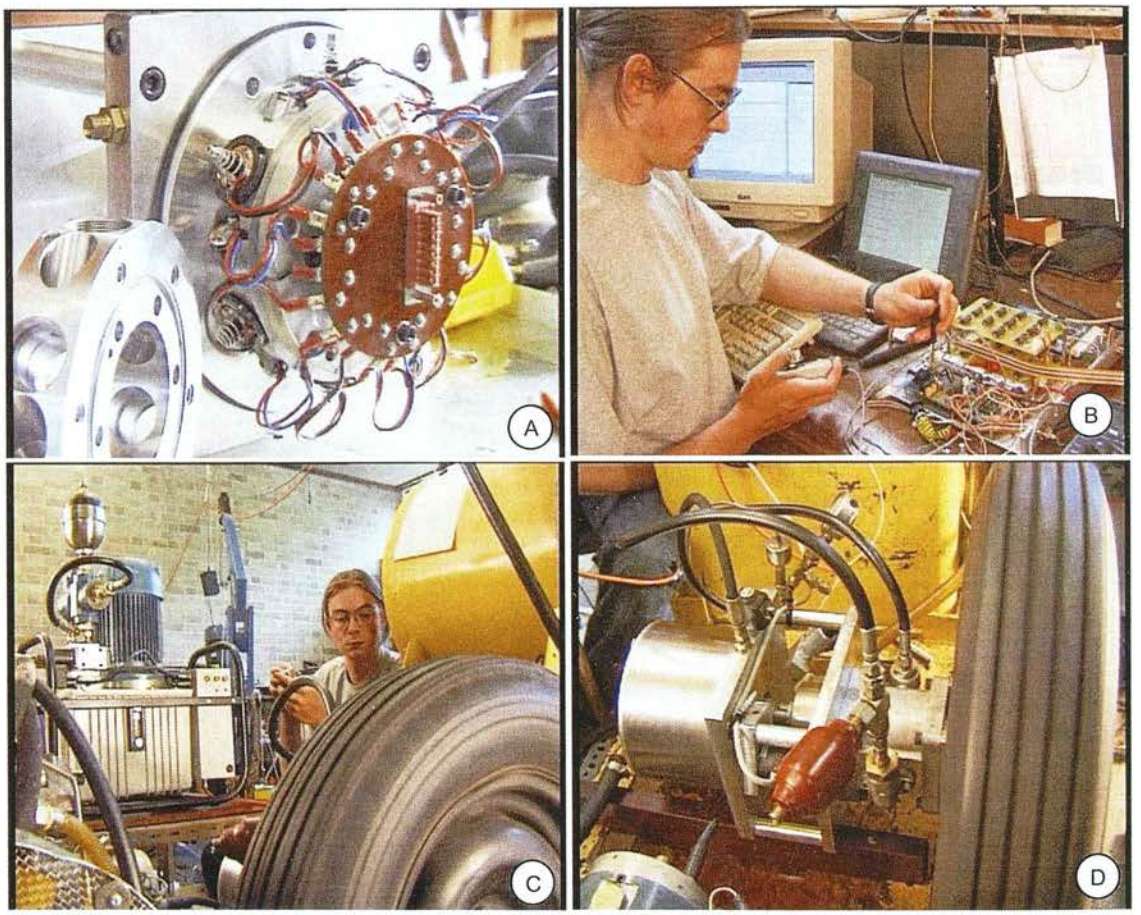


Figure 4-17: Early propel DDPM development in the lab (from the video "Power for Change", courtesy Jamie Taylor)

Referring to the labels above: (A) assembled DDPM with tank removed, showing coil connections and mk.1 NCSO LPVs; (B) adjusting electronic PWM oscillators; (C) motoring trials with a DDP power-pack as the supply; (D) DDPM driving load comprising gear pump working against a needle valve, and inertia in the form of a truck wheel.

4.7.2 The false rev

When the DDPM controller was first powered up, or following a hardware reset (triggered by a button), the valves were all switched off. In this state, the DDPM reverted to a passive state where the HPVs and the LPVs were closed. Due to the choice of NCSO LPVs, in this state the DDPM behaved as a check-valve pump. If there was a torque on the shaft (because the

vehicle was on a hill), then the DDPM would work passively as a pump at full displacement, pushing fluid into the high-pressure port. No fluid can flow back into the disabled supply DDP which was connected to this port, so the accumulator pressure would rise, and the DDPM created torque that resisted the tendency of the vehicle to roll down hill. In this passive (or "idle") state, the vehicle would slowly creep down hill, at a speed determined by the DDPM internal leakage at the pressure needed to resist movement. On a hill requiring 100 bar of holding pressure, this creep speed was typically 2rpm, or 0.11% of the maximum speed of 1800rpm, indicating a very low level of internal leakage.

In the early development, the DDPM was commanded by a potentiometer to one of three states. In the mid point of this pot, the idle state was commanded. If the pot signal went above a dead-band, the "forward" state was commanded. If the pot went below the dead-band, the "reverse" state was commanded.

On transition to the "forward" state, the first action of the controller was to measure the position of the shaft so that the commutating valves could be set to the correct state. The value of the DAC was sampled, and the position of the shaft determined. The angular position of the eccentric was defined relative to BDC of cylinder no.6 as shown below:

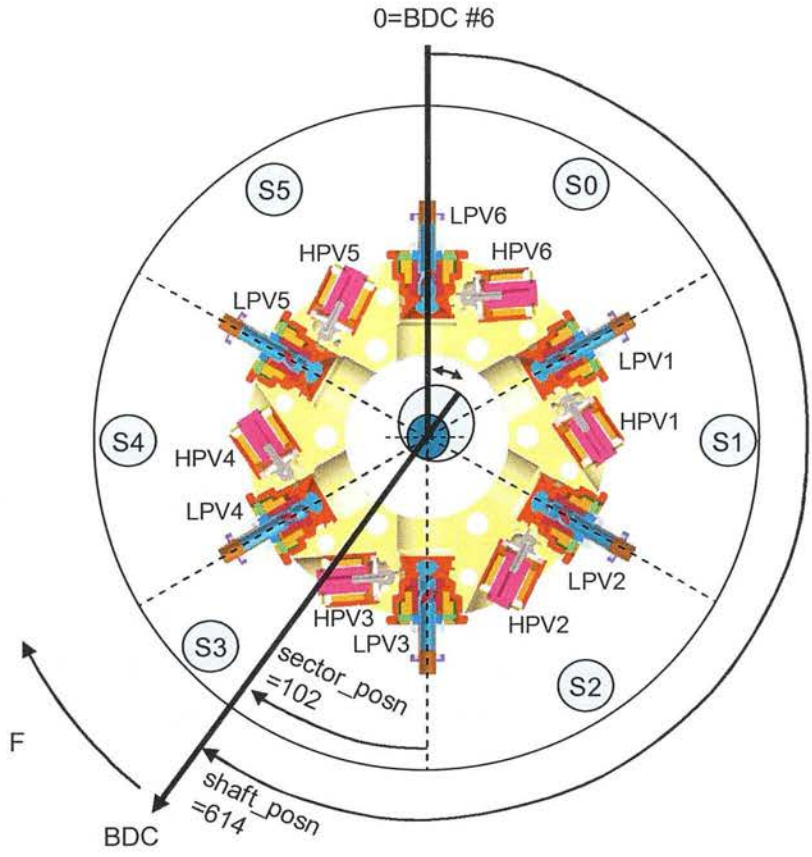


Figure 4-18: Position measurement at start of false rev, going forwards

The voltage from the DAC was measured as a 10-bit integer as the variable "shaft_posn". The encoder was carefully synchronised such that shaft_posn=0 corresponded to BDC of cylinder #6. The 360 degrees of angular position were split into 6 sectors, corresponding to the 6 cylinders. The position in the current sector was calculated as in the figure above and set the variable "sector_posn". The calculations for the "forward" mode are shown above; those for the "reverse" mode were mirrored. An array of integers held the sequence of high and low pressure valve events needed to complete a revolution, and from these arrays and the sector calculated above, the correct state of each HPV and LPV was determined.

Once the position was measured, the false rev sequence began:

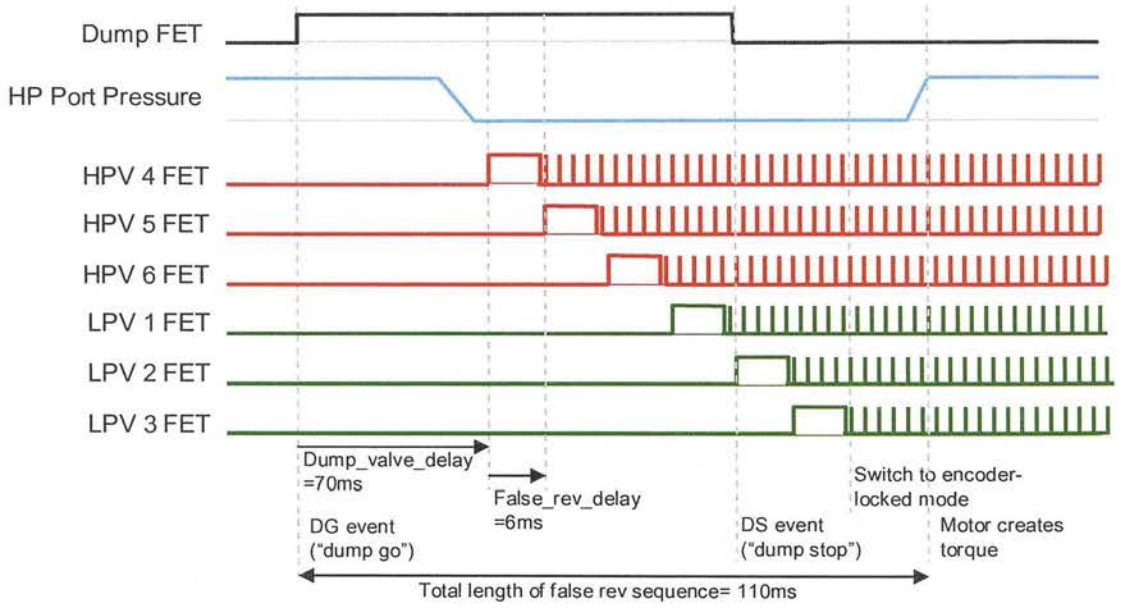


Figure 4-19: "False rev" sequence

The first event of the false rev sequence was the start of the dump valve FET pulse. The dump valve had been carefully calibrated, and it was seen that the pressure in the high pressure port of the DDPM was completely exhausted after 70ms at 200 bar. The majority of this time was a delay caused by electrical and mechanical time constants of the valve; once the valve was open, the depressurisation took only 15ms.

After 70ms had elapsed, the valve firing sequence began. The valves were fired in the order they would be were the motor rotating in the direction demanded, starting from the measured position, at a separation of 6ms - corresponding to a shaft speed of 1670rpm. Because the shaft was in fact stationary at this time, the sequence was known as the "false revolution" or "false rev". The 6ms separation was important to make sure that valve pulses did not overlap. Experiments showed that if they did overlap, the voltage drop in the cables would cause the

voltage across the coils to deviate from the nominal 12V, causing the actuation time of the valves to exceed the FET pulse duration.

The dump valve was de-energised before the final valve was fired, because calibration tests showed that pressure started to rise 65ms after the valve was de-energised. This ensured that the overall time spent de-pressurised was as short as possible.

At 6ms after the final valve was fired, the controller switched from a clock-based increment of the interrupt register to an encoder-based increment. Interrupts were set on the value of this register for the next LG, LS, HG and HS events due in the motoring cycle as the shaft rotated.

4.7.3 Position-locked motoring operation

As the shaft rotated, the controller tracked the position of the shaft by incrementing the interrupt register on every pulse of the LSB of the encoder signal. As this register incremented, an interrupt would be triggered when the shaft position was equal to the position at which each event was due. In between valve events, the embedded software was in an idle mode, waiting for the next interrupt to occur.

For each revolution of the motor, there were 24 events; 4 for each of the cylinders, as shown below:

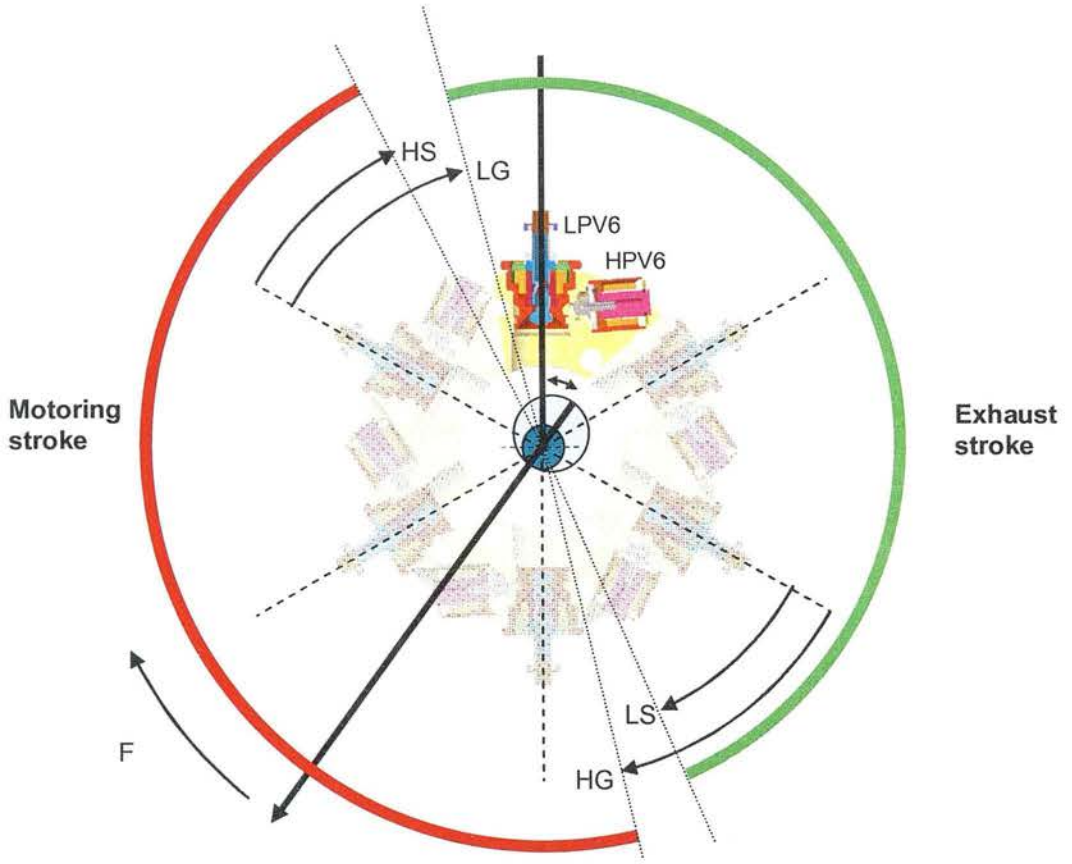


Figure 4-20: Valve events for cylinder #6 as a function of shaft position

For clarity, the diagram above shows the sequence of valve events for just one of the six cylinders. As each cylinder was 60° out of phase, there were always one event of each type (LG, LS, HG and HS) in any one sector, for a total of 24 events per revolution. Each valve event position was defined relative to the start of the sector in which it took place as shown above; a full revolution was divided into 10 bits (=1024 counts), so each sector spanned 170 counts. So the valve event positions could take a value between 0 and 170.

The rising edge of the encoder MSB triggered a reset of the position register. At this point the interrupts were set for the next due events of type HG, HS, LG and LS. The position register was incremented by the encoder LSB as the shaft rotated, and these interrupts were triggered at the position pre-set at the time of the MSB edge. When the first event of each type was executed, another interrupt was pre-set for one-sixth of a revolution (i.e. 170 counts). The first event of each type in a revolution was explicitly placed at an absolute position by the controller, but the subsequent five events of that type were simply spaced 60 degrees from the previous event.

The DDPM cycle was fundamentally unaltered compared to the DDPM cycle shown in Section 3, however the NCSO LPVs were controlled in the negative sense compared to the

fixed-speed DDPM, which used NOSC LPVs, i.e. the LG event was triggered whenever the valve was to open, and the LS event was triggered whenever the valve was to shut.

Initially valve events were triggered at fixed shaft positions. There was no compensation for the time delay in between firing the FET and the actual valve physical event completing. Attention was concentrated on achieving a stable motoring cycle, with excessive pressurisation and depressurisation time, such that the HPV was always opened repeatably at TDC. Once this was achieved, the position of the HS and LS events were increased (i.e. they were made to occur later and later in the cycle) until the motoring cycle started to fail, then decreased slightly to provide a safety margin. This established the HG, HG, LG and LS positions for low speed operation. A supply pressure of 100 bar was used for all of these tests.

As might be expected, fixed valve positions only allowed stable operation up to about 300rpm. To achieve higher speeds, the time delays for the valve events to complete their physical state changes (i.e. open to closed, or closed to open) would have to be allowed for, by advancing the event positions as speed increased. As each type of event had a characteristic time delay, each of the event types would have to be advanced at a different rate as speed increased. This is analogous to the centrifugal timing advance mechanism built into an engine ignition distributor.

The speed of the DDPM was calculated once per revolution by measuring the period between successive rising edges of the encoder MSB. The position advance function operated on the first event of each type after the encoder MSB. Subsequent events were spaced at exactly 60 degrees apart until the next MSB. Therefore the speed-relating advance was only updated once per revolution. This is acceptable if it is assumed that the DDPM is always driving an inertial load like a vehicle which limits the maximum shaft speed rate-of-change. Even with only the truck wheel as the inertial load, which could reach full speed in less than one second, no adverse effects from this once-per-rev update were noticed.

The position advance (measured in counts of the 10-bit encoder) required of an event to compensate for a fixed time delay, is proportional to the speed. In the case of the LG event, the position at low speed was 68, while the position required at 1500rpm was 14 (arrived at by experiment). Thus the total advance required was 68-14=54 counts. For each valve event type, a divisor of the speed (in rpm) was derived by experiment, whereby advance = rpm/divisor. These are summarised in the table and graph below:

Event	Event position, low speed	Event position, 1500rpm	Rpm Divisor	Equivalent time, ms
HG	150	15	25	2.4
HS	105	-2	9	6.5
LG	142	14	28	2.2
LS	100	60	15	4

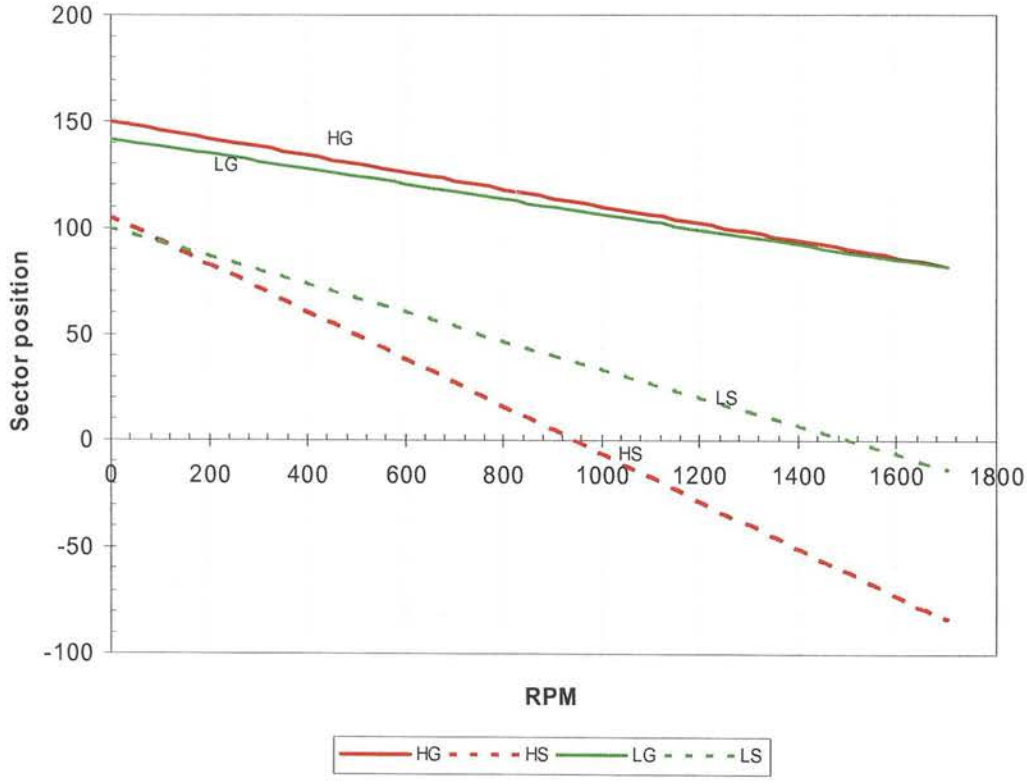


Figure 4-21: Valve event position advance as a function of speed

Note that the HS and LS event positions become negative at high speed. Code was written so that events were able to advance into the preceding sector as necessary when this occurred so that position advance was continuous with speed.

The delay times for the HS and LS events are significantly longer than the HG and LG events. Even with ideal springs, the total release time for both valves was longer than the total actuation time. The reasons for this was that the duty cycle of the pulsed part of the valve sequence had to be increased significantly from the values which achieved stable latching on the bench, where the only force acting to release the valve was the return spring. The flow

forces in the working machine increased the required latching force, and therefore the required pulse duty cycle.

Once the motoring cycle was operating, attention turned to the pumping and idling cycles.

4.7.4 Pumping cycles

The pumping and idling cycles are simpler than the motoring cycles, as only the LPV needs to be actively controlled.

Although the machine was designed to operate passively as a check-ball pump, at high speed the LPV would not open far enough from suction alone to prevent cavitation and produce full effective displacement. This is because the return spring of the LPV produced a force which exceeded 1 bar of suction pressure after the valve had travelled only 0.6mm of the total 1.5mm travel. Therefore an active pumping cycle was implemented. In contrast to the fixed speed DDPM, which used NOSC LPVs, the propel DDPM used NCSO LPVs, so the valve had to be released before BDC, and latched before TDC:

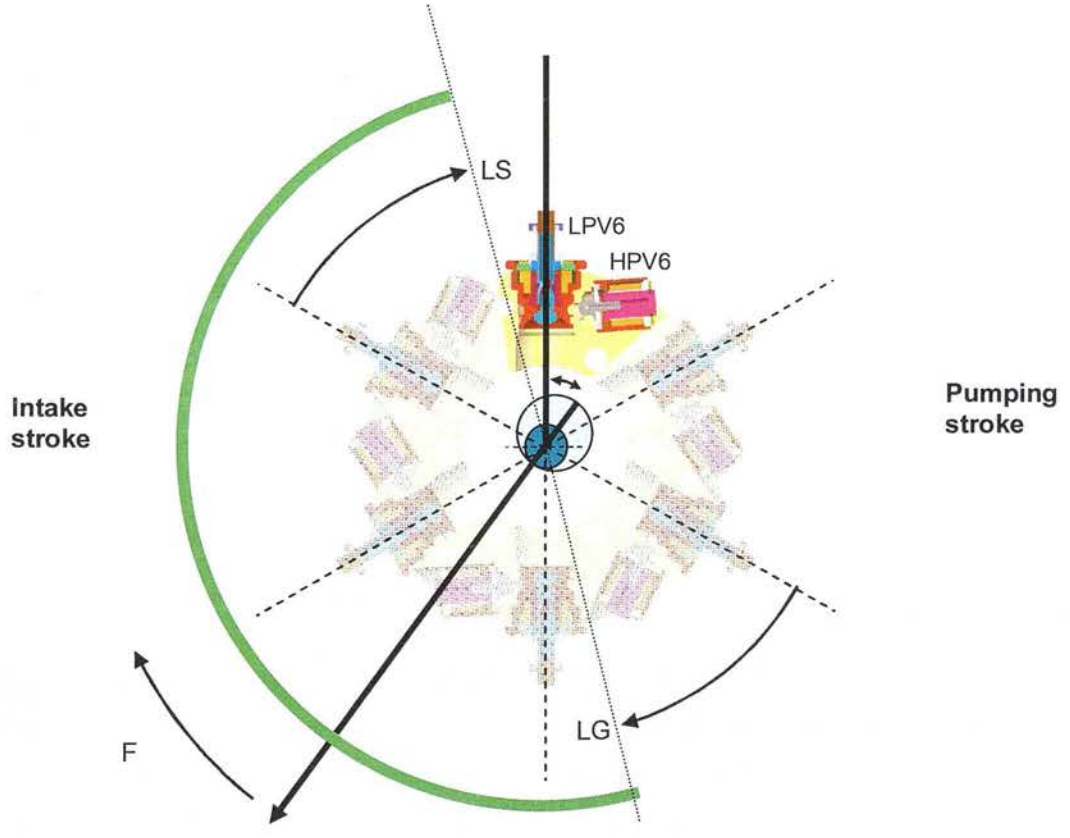
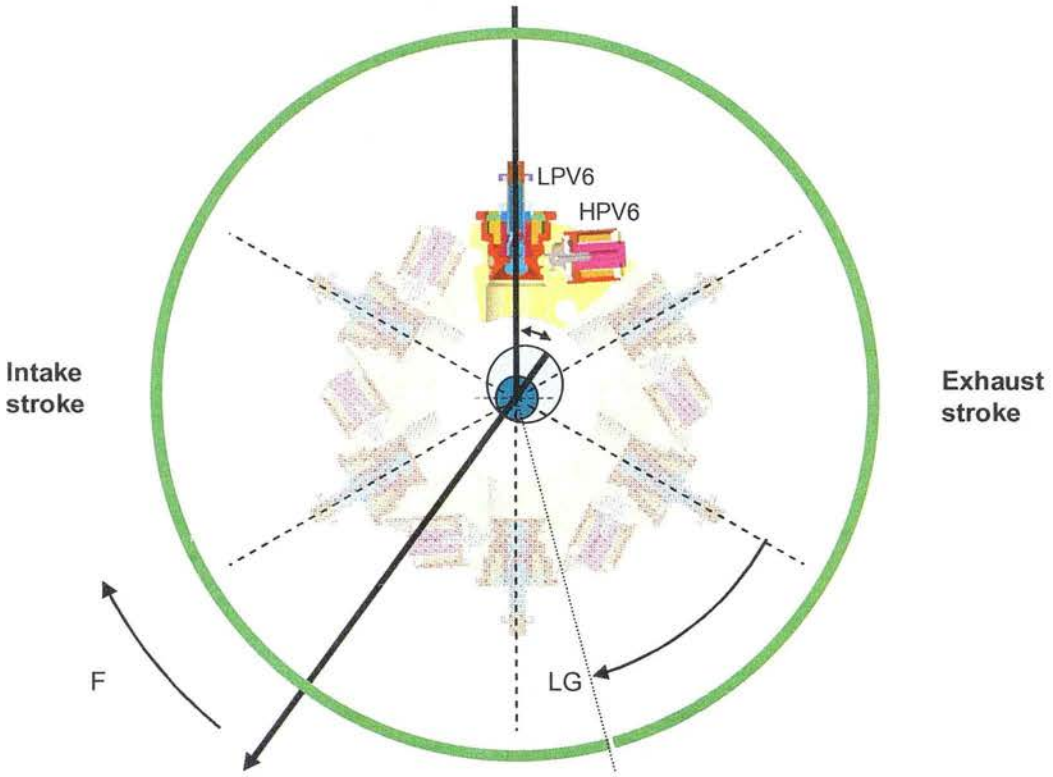


Figure 4-22: Valve events for the pumping cycle

4.7.5 Idling cycle

In order to achieve variable displacement by cylinder disabling, the DDPM must be able to execute idling cycles. With NCSO LPVs, such a cycle consists of latching the LPV before TDC and keeping it latched for the entire duration of the revolution. If the cylinder is to execute a continuous sequence of idling strokes, it would be possible to dispense with the initial pulse in all subsequent cycles as the valve is already latched, to save electrical energy; this was not implemented in practise to provide a degree of robustness; if the valve fell shut for whatever reason (for instance, due to insufficient latching current) then it would be actively re-opened at every TDC. Hence the initial pulse was re-triggered every revolution in the cycle shown below:



4.7.6 Variable displacement

Once the low-level valve control code was developed, the higher-level control layers were added. To achieve variable displacement, the propel DDPM needed to be able to mix idle strokes with motoring strokes, and idle strokes with pumping strokes. This was achieved through a simple accounting algorithm as described by Rampen (1992) as the "flow" algorithm (more accurately the "displacement" algorithm, as it does not account for variations in shaft speed). Six times per revolution, there is a decision point which coincides with the LS event, chosen because it is common to both pumping and motoring cycles. At each decision point, the register *dec* was incremented with an integer *torque_err* (-100 to 100). If the value

of *dec* exceeded 100, then a motoring cycle was initiated and *dec* was decremented by 100. If the value of *dec* was below -100, then a pumping cycle was initiated and *dec* was incremented by 100. If neither a pumping nor a motoring cycle was initiated on that decision point, then an idling cycle took place. In this way the time-averaged motor displacement could be continuously varied between -100% (pumping) and +100% (motoring), although the instantaneous displacement depended on the combination of cylinders which were enabled at any one time. The sequence of decisions resulting from this algorithm is termed the enabling sequence, as shown in the figure below:

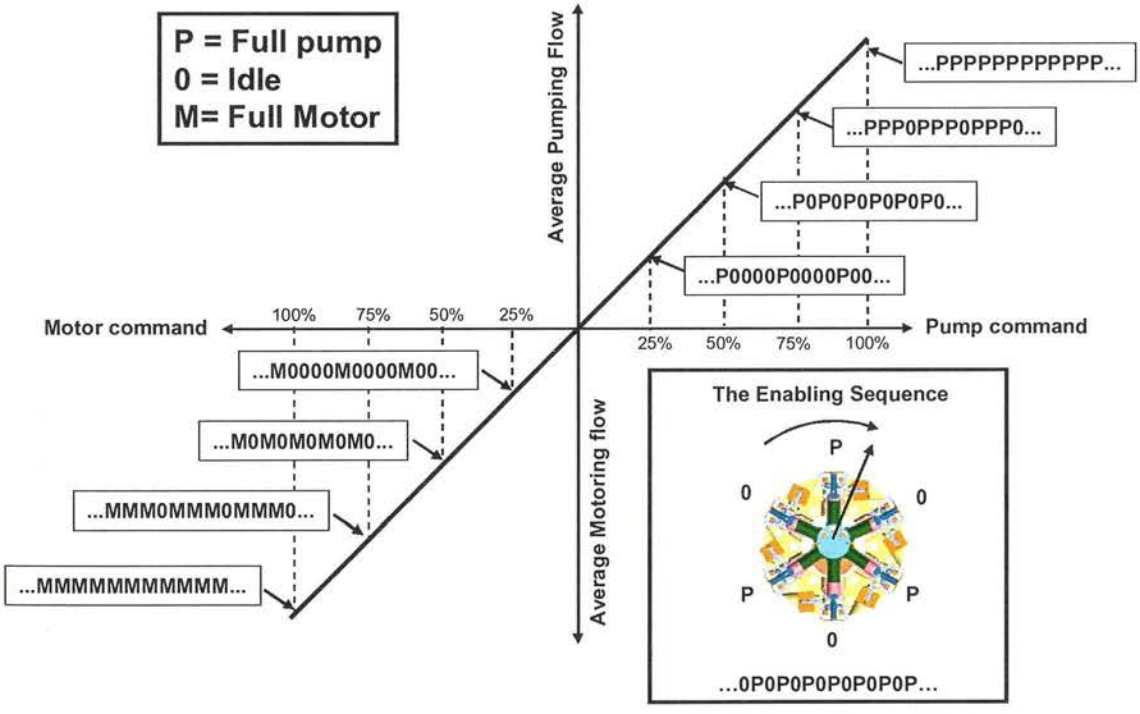


Figure 4-23: Continuous displacement control of the propel DDPM

4.7.7 Secondary speed control

Once the motor had displacement control, secondary speed control was investigated. A simple PI controller was implemented, with a potentiometer providing a speed set-point. The speed control loop was executed once per revolution, after the speed had been measured. The control function was:

Displacement(%) = *SpeedError**2 + *IntegError*/100

where *SpeedError* was the error in rpm between the set-point and the actual speed (a positive error meaning that the actual speed was below the set-point);

IntegError was the result of accumulating (3**SpeedError*) once per revolution;

and *Displacement*(%) was limited to +/-100

The integration function was executed one per revolution, rather than on a continuous free-running timer, so the effective integration gain was the first power of speed. This makes some intuitive sense, because the control bandwidth of the DDPM increases with the first power of speed. This amounts to a simple form of gain scheduling.

For these tests the pressure was 100 bar, maintained by the DDP pressure control loop. The needle valve on the gear pump load was set so that the steady –state speed of the DDPM shaft at full displacement was 700 rpm. Below that speed, the needle valve would create a lower pressure on the gear pump, so the torque on the shaft would be lower.

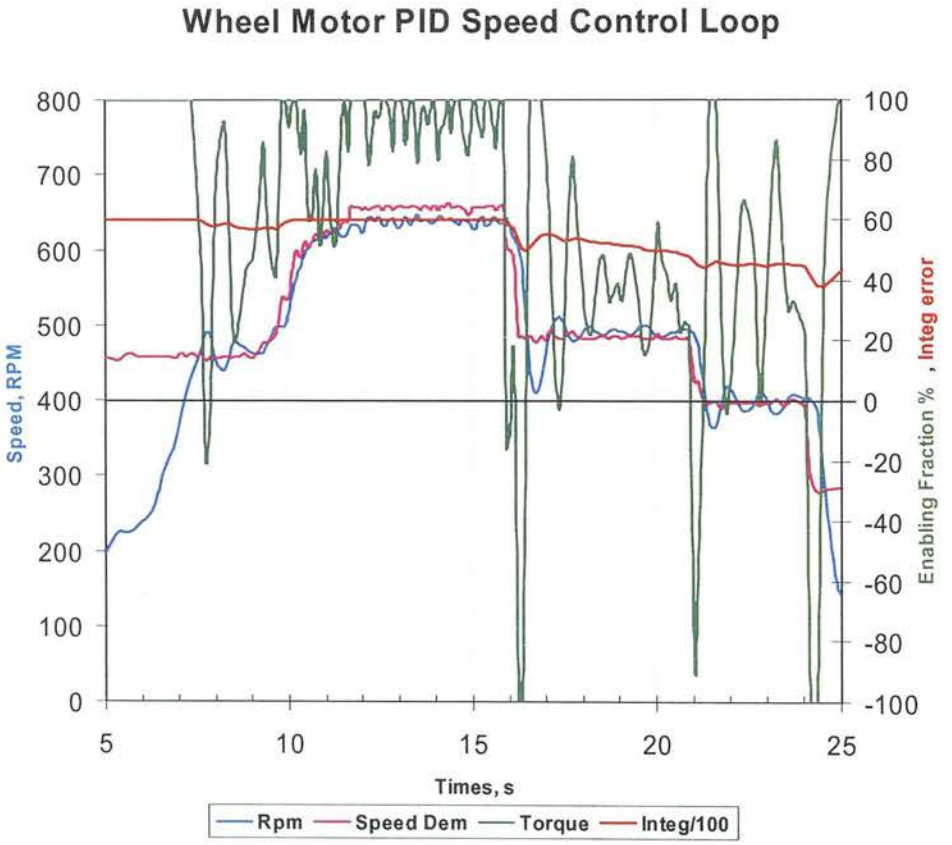


Figure 4-24: Secondary speed control of propel DDPM

The graph above shows that in principle, control of DDPM shaft speed control by secondary control of DDPM displacement has been demonstrated. The actual motor speed stabilised close to the demanded speed after each speed demand step, exhibiting the classical overshoot behaviour of a PI controller. However, there was no systematic study of the limits of stability or performance of such a control loop.

4.7.8 Electronic load-sensing control

The above result shows speed control by secondary control of DDPM displacement, with a constant pressure set-point. This is typical of secondary control applications with a large energy-storage accumulator (such as an automotive parallel hybrid transmission system), where the system pressure is determined by the state of charge of the accumulator (Jen and Lee 1993).

The introduction of pump pressure set-point as a degree of freedom, requires that a strategy be defined as to what the steady-state of the system should be for a given motor speed and load. If the system comprises a DDP supplying a propel DDPM, that decision could be made on the basis of the optimum operating point of the system from the point of view of energy losses, for instance, keeping the pressure high so that the flow was kept low, minimising flow-related pressure losses and allowing the engine to work at a lower speed where it was more efficient.

However, in many mobile applications currently served by open-circuit propel systems, the working functions are also supplied by the same pump which would supply the propel DDPM.

Pressure-controlled swashplate pumps are commonly used in mobile applications as the pressure source for a bank of load-sensing, flow-compensated proportional valves (Krus 1991, Bauer 2005). Each valve in the bank measures the load pressure required by means of a shuttle valve, and a network of check valves communicates the highest of these pressures back to the pump as a load-sense pressure. The mechanical pressure compensator in the pump controls the swashplate such that the outlet is maintained at a certain margin above the load-sense pressure. This pressure margin must be sufficient to overcome pressure drop inside each valve itself, plus the pressure drop in the line between the pump and the valve stack. Typically this margin is in the range 10 bar to 30 bar.

In the future it is to be expected that the mechanical controls in both the pump and the valves will be replaced by electronic ones. In such a system, the shuttle valves and check valve network is replaced by pressure transducers built into each valve, each of which will have a local microcontroller. The load-sense hose is replaced by a digital network such as CAN bus. Such an "electronic load-sensing" system offers significant advantages, such as software-variable pressure margin for higher energy efficiency, and bi-directional communication between valves and pump controllers for better utilisation of pump capacity. The DDP with a pressure-control loop would offer significant advantages in this role, because of better part-load efficiency, and the ease with which higher-level pump control strategies, such as power-limiting and anti-stall (Paoluzzi et al. 1996), can override the pressure-control loop.

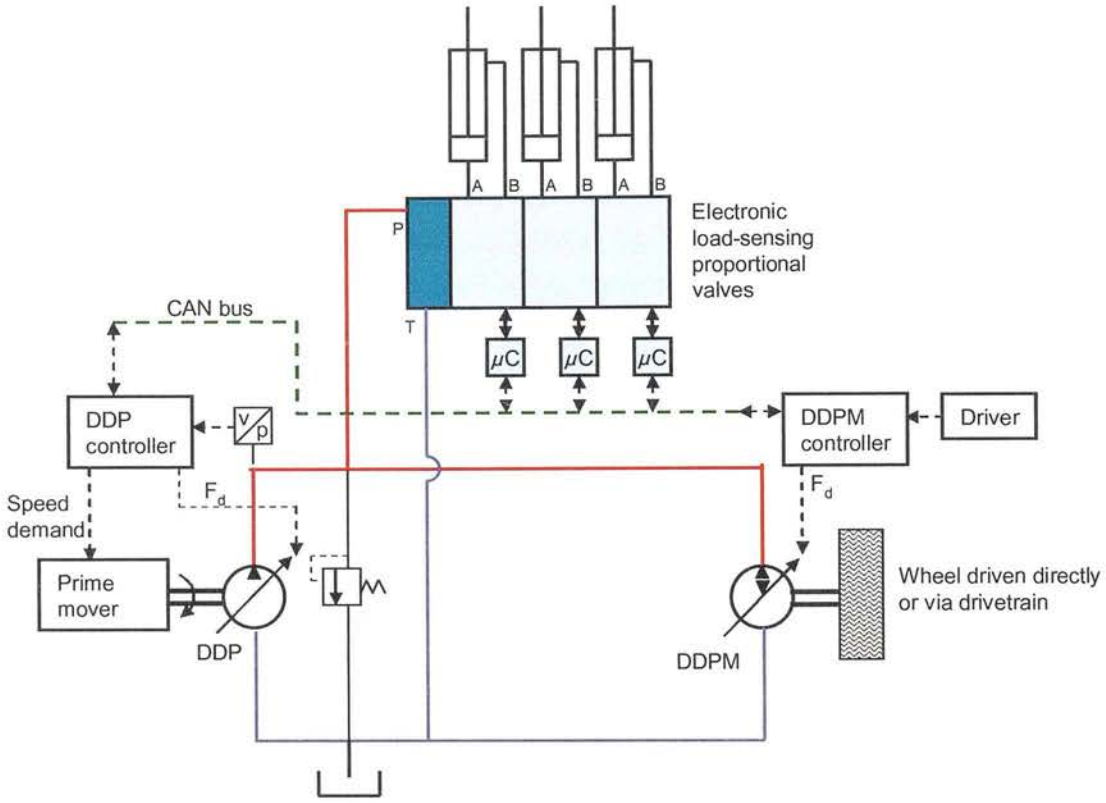


Figure 4-25: Concept for secondary control of a propel DDPM in an electronic load-sensing system

The concept pursued was to consider the DDPM as being similar to a flow-compensated proportional valve in a load-sensing system. At the basic level, the DDPM controller would simply need to send back to the pump controller a demand pressure signal like the proportional valves. However the bi-directional signal path would also allow other signals to be exchanged:

- The DDP pressure control loop could benefit from a signal corresponding to the flow being taken by the DDPM, for use as a feed-forward term to the DDP pressure control loop.
- The DDPM could receive a request from the DDP to limit the displacement of the DDPM, for instance if the flow limit of the DDP was about to be reached, or if a demand was suddenly made of a working function which had a higher priority than the propel function.

For the demonstration vehicle, the DDPM was controlled as if it were one consumer in an electronic load-sensing system.

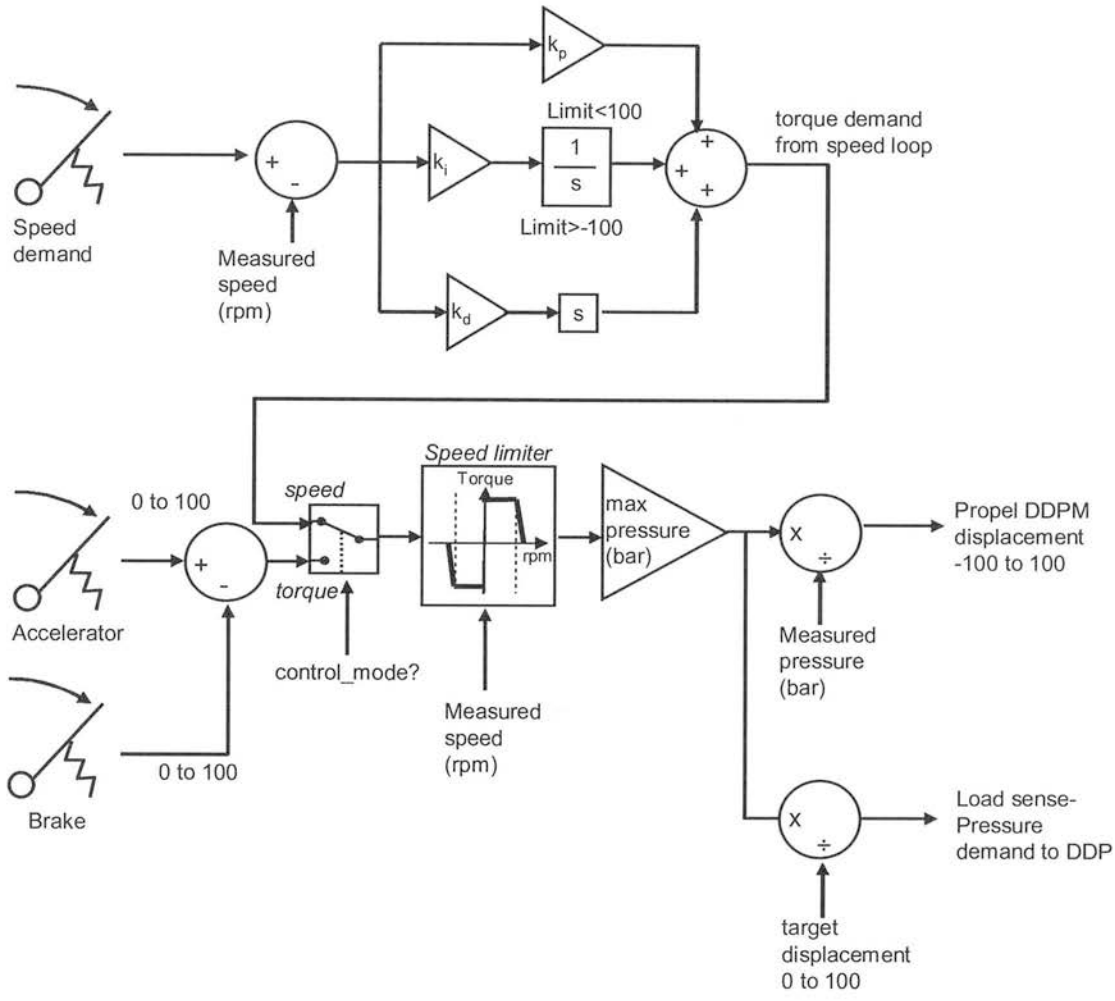


Figure 4-26: Control algorithm for the propel DDPM vehicle

The above diagram shows the algorithm used for the propel DDPM vehicle. Note that the differentiator and integrator blocks are described by their equivalent Laplace tranforms, although they were implemented as discrete functions, as the controller worked with a fixed sample rate The earlier speed control algorithm was triggered once per revolution of the DDPM. For vehicle work a time-based interrupt was required, to prevent the controller sample rate dropping to zero when the vehicle was stationary. The control loop above was executed at 16.7Hz, this frequency being limited by the need to avoid interfering with the low-level valve events. Unlike the DDP propel system presented in Section 5, the shaft-position-based control and the time-based control were executed by the same controller, and in retrospect these asynchronous functions are better split into two separate controllers.

The accelerator and brake pedals together make a bi-polar torque demand, as does the speed control PID loop, where +100 is the maximum driving torque capable of being exerted by the DDPM at full motoring displacement and maximum pressure, and -100 the maximum braking torque at full pumping displacement and maximum pressure. The flag *control_mode*

determines whether the driver is controlling the vehicle in speed or torque control mode, and therefore which torque demand is selected. This flag was set by observing which control was first moved when the vehicle was stationary, and would stay set until the vehicle stopped again.

The torque demand is processed through a speed limiter. The function of this block is to prevent the DDPM from exceeding a set maximum speed. In the early stages of DDPM development in the lab, the machine operated up to 1800rpm. This was limited by the position advance code, which did not allow valve event positions to be advanced by more than one sector from the low speed position. However, as the higher-level functions were added, particularly the RS232 data acquisition, the load on the processor increased to the point that erratic behaviour was noticed above 1500rpm. Therefore a speed limiter function was inserted. Above the maximum speed set-point, the torque was scaled down progressively, so that by the time the speed was 200rpm above the set-point, the torque was scaled to zero. The speed limiter did not prevent over-speeding if going down a hill. However, due to the choice of NCSO LPVs, the fail-safe condition of the DDPM was to create a retarding torque. This meant that if the controller "crashed" due to over-speeding, the vehicle would safely come to a stop.

After the speed limiter, the torque demand (+/-100%) was scaled by the maximum pressure of the DDP – typically 200 bar – to give demanded torque as the product of pressure (bar) and displacement (%).

The demanded DDP pressure is equal to this product, divided by a target displacement. If the target displacement is 100%, then the pressure demand is the minimum pressure at which the motor is capable of producing the demanded torque. In a steady state, where the DDPM is the only consumer of flow, the DDP pressure will tend towards this pressure and the DDPM displacement will tend towards maximum. This is the behaviour most likely to be acceptable in a load-sensing system, because it keeps the LS pressure as low as possible and therefore minimises pressure drops in the compensators of the working functions. However if the target displacement is 50%, then the DDP will tend to work at twice the minimum pressure necessary.

If the instantaneous DDP pressure is higher than the minimum pressure required, then the DDPM works at reduced displacement such that only the demanded torque is produced. This is analogous to the pressure drop created in a load-sensing proportional valve by the flow compensator, if the pump pressure is above that demanded by that valve.

As is the case with load-sensing systems in general, this arrangement causes the dynamics of the system to be asymmetric with regards to increases and decreases of external loads.

If the load on a load-sensing valve suddenly reduces, the local flow compensator can suddenly close to create a pressure drop to keep the flow to the load constant. In due course the reduced LS pressure demand to the pump will cause the pump pressure to reduce to the new lower level and the flow compensator will open up. The valve can respond very rapidly to keep the load flow constant if there is a sudden reduction in load.

However, if the load suddenly increases, then the flow compensator opens fully but this does not necessarily increase the pressure at the load sufficiently to satisfy the demand. The LS signal from the valve increases and the pump responds by creating this pressure plus the margin pressure. The rate of change pump pressure is limited by the response speed of the pressure compensator in the pump, the length and compliance of the LS hose, and the compliance in the main pressure hose. The result is that the valve cannot respond rapidly to keep the load flow constant if there is a sudden increase in load.

This is the same behaviour seen with the DDPM in a load-sensing arrangement. If the target displacement is 100%, the DDPM can very rapidly reduce displacement to satisfy reduced load, but must wait for pump pressure to rise to satisfy increases in load which are greater than the margin pressure. If the target displacement is 50%, then the pump will be at higher pressure all the time, and the DDPM has the ability to very rapidly increase and decrease torque without having to wait for the pump to increase pressure. Therefore, in an application where fast response is necessary, it may desirable to work with a target displacement less than 100%, or a higher pump margin pressure. In a fully electronic load-sensing system, both the target displacement and the margin pressure may be varied continuously by an overall vehicle controller in response to how the operator is using the machine.

The first results from this algorithm show that the combination of a target displacement of 100% and a minimal margin pressure can cause asymmetric (and therefore non-linear) response. This non-linearity makes the tuning of the speed controller more challenging.

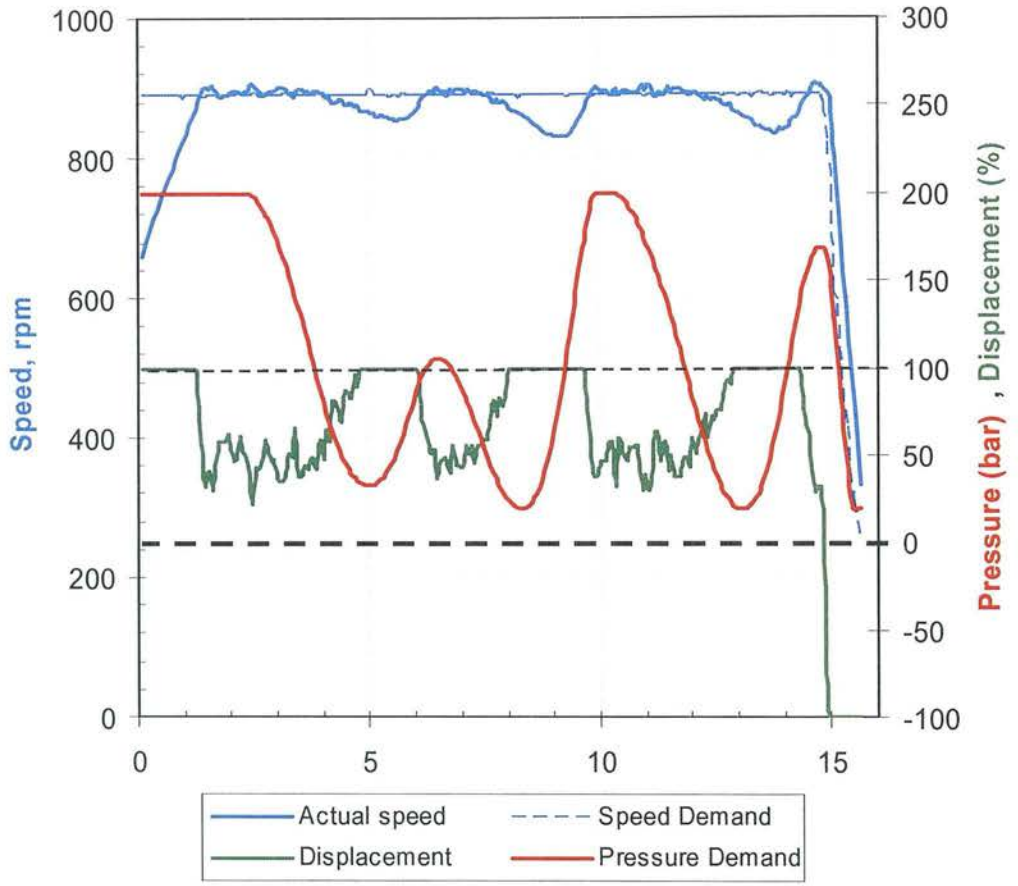


Figure 4-27: What can go wrong with an electronic load-sensing DD transmission

The above results shows what went wrong with the first attempt at implementing a DDP/DDPM electronic load-sensing transmission system with secondary speed control. At t=0, the DDPM is well below the speed set-point so the pressure demand and displacement are at maximum. As the speed reaches the set-point, the displacement is modulated by the speed controller to maintain the demanded speed. As the displacement required is only 50%, the pressure demand falls. However the interaction of the pump time-constant and the DDPM speed controller dynamics cause a phase lag which results in the pressure dropping below the minimum required at t=5. The DDPM displacement saturates at 100% and the speed starts to recover. However by this time the speed error is sufficient that the integrator produces an overshoot to the pressure demand. This limit cycle repeats.

This behaviour was much improved by use of a margin pressure of 20 bar. In the steady state, this kept the DDPM displacement slightly below 100% at all times, and therefore in the region of high-bandwidth control.

When braking, the DDPM pumps fluid into the high-pressure line. The DDP controlling the system pressure has no ability to motor this fluid back to tank, so the tendency is for the system pressure to rise to the relief valve setting during braking, such that by the time the vehicle is stopped, the system pressure is at maximum. If the braking torque demand is low, this leads the DDPM to run at very small fraction of full displacement. Under the conditions of low speed, low enabling fraction and high pressure, the sparse torque pulses from the DDPM could be felt by the driver.

In a multi-consumer load-sensing system it is possible that some of this braking energy could be used by any working functions which happened to be active at the time which would help to limit the pressure rise. However in the demonstration vehicle there were no other consumers, so a solenoid-actuated normally-closed 2-port 2-position valve was used to vent the high pressure fluid to tank via an orifice. A simple 'bang-bang' control algorithm in the DDPM controller kept the system pressure below 20 bar above the pressure demand.

At this time, the "part stroke" algorithm described in Section 5.3.1 had not been developed, and the DDPM could only execute full motoring strokes. It is thought that a DDPM with part stroke motor and pumping cycles will not suffer from perceived torque pulsation at low speed, high pressure and low displacement, but this has not been demonstrated.

It should be noted that even with the speed control algorithm working perfectly, the actual DDPM speed can never track the desired speed perfectly. This is because the DDPM delivers torque in pulses, and therefore the DDPM speed will always waver around the set-point by an amount inversely related to the inertia of the load. This is very similar to pressure control in a DDP, which will always lead to a steady-state pressure ripple inversely related to the compliance of the load. With the proportional gain set to achieve good transient response, the result can be that the controller tries to counteract the ripple inherent in the DDP/DDPM concept, leading to chaotic results.

In both cases it is the experience of the author that the stability of the control loop is helped greatly by introducing a band of reduced proportional gain, of the same order as the expected steady-state ripple, into the proportional part of the PID algorithm.

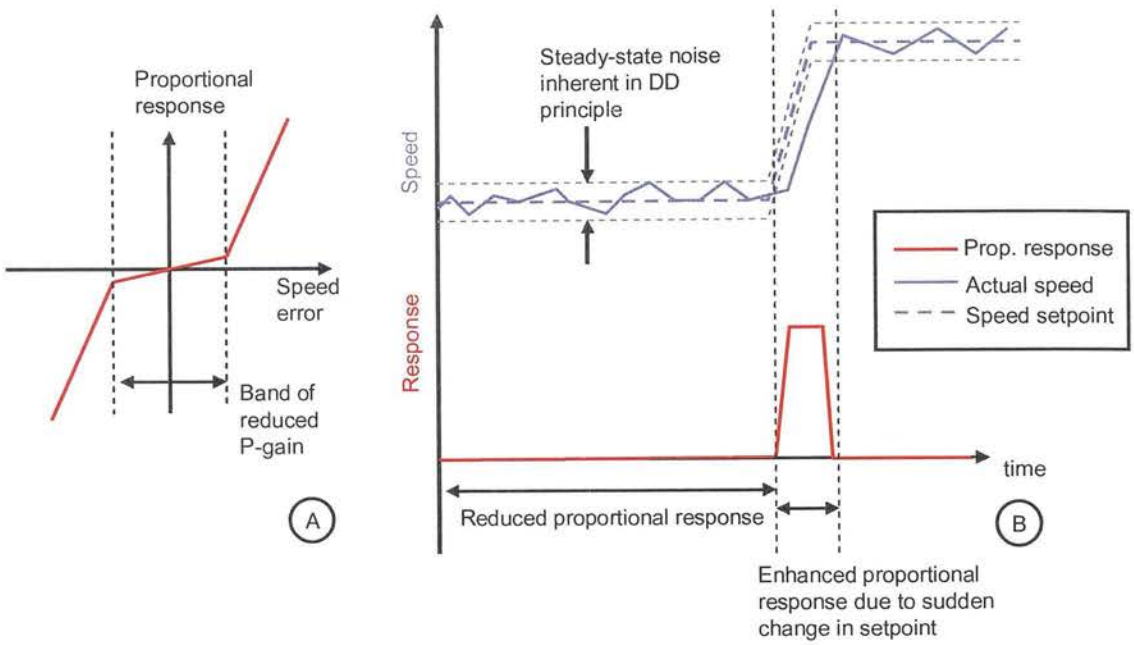


Figure 4-28: "Variable structure control" applied to DDPM speed control

This technique allows the proportional gain outside of this band to be much higher than it otherwise could be, greatly improving transient response without over-controlling the displacement at steady-state conditions. Switching gains as a result of the observed state is a simple example of a technique classified by control theoreticians as "Variable Structure Control" (DeCarlo et al. 1988).

Much work remains to be done on DDPM control algorithms. The response of a DDPM to a control input, the synthesis of discrete half-wave sinusoids, phase-locked with the shaft, is fairy unusual in the mechanical engineering world. However in the electrical world there are close parallels with a class of power electronic converters using Silicon Controlled Rectifiers classed as "zero-fired", or "zero voltage-switched". There is also some similarity with the operation of sigma-delta analogue-to-digital converters used in digital audio (Pohlmann 1995). The author expects that control techniques from the electronic world could usefully be applied to the DDPM control problem in the future.

4.8 Vehicle demonstrator

The propel DDPM and a DDP were packaged into a golf buggy vehicle, powered by a petrol engine. The details of the DDP mounting to the engine, the belt drive and the instrumentation are covered in Section 5.2.2. The differential was driven by the propel DDPM shaft by a belt drive as shown below – the overall gearing gave the vehicle a speed of 16km/hr at a DDPM speed of 1500rpm.

The hydraulic circuit was as described previously. A 300cc accumulator (pre-charge=20 bar) was connected to the high-pressure line.

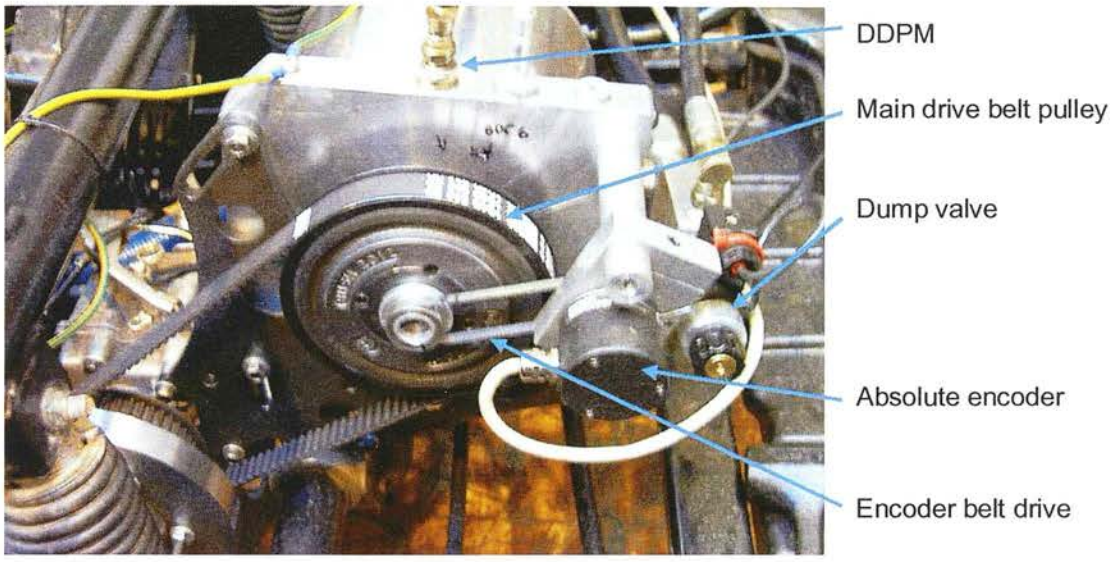


Figure 4-29: The propel DDPM installed in the demonstrator vehicle, showing the absolute encoder, synchronised to the DDPM shaft with a small toothed belt.

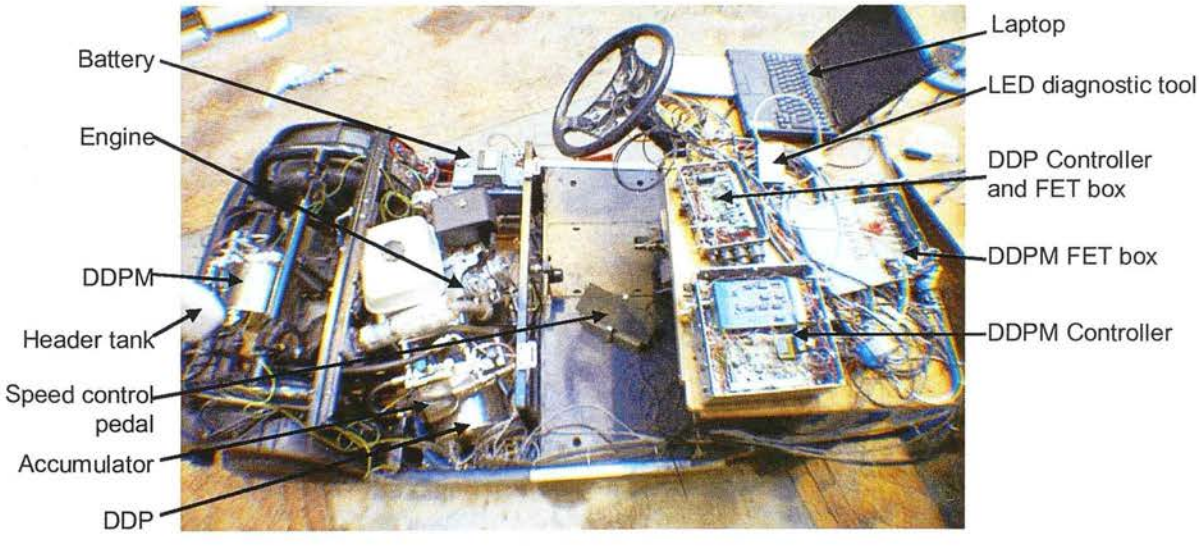


Figure 4-30: Layout of the demonstrator vehicle during development

4.9 Vehicle testing and development

4.9.1 Diagnostics

The development of a complex system is only possible if the details of each component can be somehow simplified inside a "black box". As the vehicle demonstrator was being commissioned, it was often unclear whether problems were the result of a failure of higher-level control, or a lower-level failure of a DDPM component. Once the DDPM was in the vehicle it was time-consuming to disassemble it so that each valve could be tested on the bench. Two special diagnostic modes were developed to verify low-level DDPM functionality, both of which were triggered by the operator:

The "click" test. The DDPM controller switched each valve on and off in a regular sequence. By looking at the LED diagnostic tool, listening for the "clicks" of the valves, and looking at the current with a clamp meter, it could be verified that the valves were being energised and were moving according to the controller's command. This test would quickly trap electronic and wiring problems but did not prove that the valves were functional.

The semi-automatic test. The DDP flow limit was set to 5l/min and the DDPM controller executed a diagnostic sequence for each cylinder in turn:

1. DDP pressure demand to zero
2. Dump valve on
3. Latch LPV
4. Latch HPV
5. Dump valve off
6. DDP pressure demand = 50 bar
7. Check actual pressure.

   If both valves were working properly, then the pressure measured at this point should be low; below 5 bar. Both HPV and LPV should be open with 5l/min flowing through from HP gallery to tank. If either valve is not open, then 50 bar will be measured.
8. Close LP valve
9. Check actual pressure

   The pressure at this point should be 50 bar. If it is lower, then a broken LPV is indicated. This check was made after experiences with LPV poppets breaking.
10. Close HP valve

11. Repeat with next cylinder

This test narrowed any problem down to a particular valve on particular cylinder.

4.9.2 Data acquisition

All of the data for the propel DDPM development came from an RS232 serial data link between the Intel 80196 controlling the DDPM, and the laptop PC. The details of this interface are given in Section 5.3.3. Problems were encountered achieving a high sample rate of this data. The sending of RS232 data occupied the processor, and unless this was scheduled carefully the crucial valve events could be missed or miss-timed. Although the main vehicle control loop executed at 16.6 Hz, the data was sent only every two samples, leading to a data collection rate of 8.33Hz. Each frame contained the following data:

Speed demand, Actual speed, Integration error, Differential error, DDPM Displacement %, Overall torque demand (+/- 20000), Demand pressure to DDP, Actual pressure

It would have been much better to supplement this RS232 data with a separate high-frequency data acquisition unit sampling the signals directly, but unfortunately this was not available to the author at the time the tests were conducted.

4.9.3 Starting problems

Every time the DDPM starts with a 'false rev', it must be disconnected from the pressure line for a short period while the valves are set into their starting position. During this time the DDPM cannot exert torque on the shaft so cannot stop it from accelerating. By the time the 'false rev' is finished and pressure is returned to the HP port, the shaft is no longer in the position that it was when the controller decided the state of the LPVs and HPVs. Therefore the DDPM cannot exert full torque on the shaft and in the worst case may actually exert a torque in the opposite direction.

Efforts were made to minimise the time spent de-pressurised, but were limited by the fact that the delay times of the dump valve were not consistent, requiring that safety margins be included. The shortest achievable de-pressurise phase of the 'false rev' sequence was 70ms.

With loads on the motor approaching the maximum torque which it could produce, the starting became more and more unpredictable. When it failed, the DDPM was programmed to try immediately to start again, and on steep hills it would often need 3 or 4 attempts, taking almost a second, before the vehicle started to move. By this time the vehicle may have rolled back by 200mm or more.

The steepest hill that the vehicle could be expected to start on is one that required 200 bar at full DDPM displacement. Given that the vehicle weighs 350kg, and is geared to do 16kph at

1500rpm, a simple calculation showed that the shaft angular acceleration during the de-pressurised phase should be 145 rad s^{-2}. At this rate, during the 70ms spent in free-fall, the DDPM shaft should rotate by an angular displacement of 0.35 radians, or 20 degrees. However, the shaft will not stop rotating the instant that the pressure returns to the DDPM. By this time the vehicle has acquired a velocity that must be decelerated before the vehicle comes to a stop and is driven up the hill. The steeper the hill, the less the margin the DDPM has above the gravitational load to accelerate the vehicle. It is not difficult to see how the vehicle could easily roll back much further than 20 degrees before stopping and accelerating up the hill.

The effect was exacerbated by the inability of the controller to measure the roll-back that occurred during the de-pressurised period. An attempt was made to cure this by sampling the DAC value corresponding to the absolute position, at a rate of 100Hz, whenever the shaft speed was below a set threshold. This gave the controller an updated position after the 'false rev' was completed, but the controller then had the problem that the valves were in the wrong state for the actual measured position. To try to tackle this, a prediction routine was implemented which "learned" the amount of roll-back which happened the last time the DDPM tried to start, and applied that offset to the position used to set up the valve states during the next 'false rev'. Some of time this improved starting, but often the effect was counterproductive i.e. if the last roll-back was 180 degrees, the motor would start its next attempt with the valves set in anticipation, and if the roll-back was not what was anticipated, the DDPM would produce torque acting to roll the vehicle further down the hill.

It may be that with a faster dump valve and better control software, it is possible to make a DDPM that starts reliably using the "false rev" method, especially for applications where the motor never faces a significant load when starting. Propelling a vehicle is not such an application. Work that has taken place since at AIP has focussed on methods of opening the HPV without de-pressurising the HP port and so far results are encouraging that the starting performance will be acceptable for propel duty.

4.10 Results

4.10.1 Torque control mode

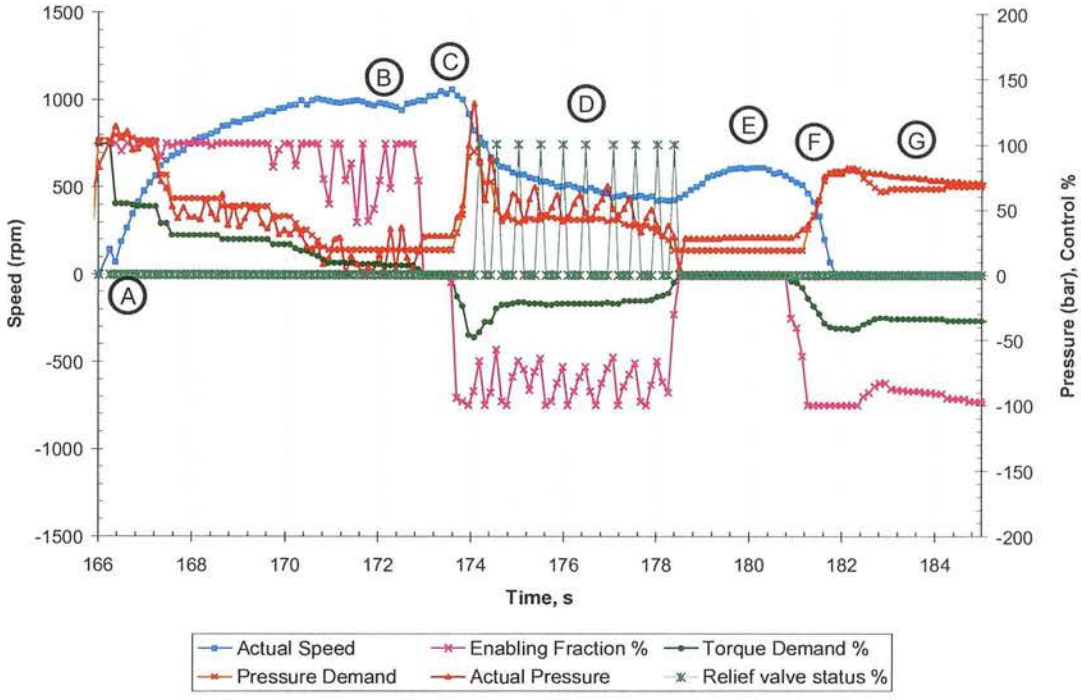


Figure 4-31: Propel DDPM vehicle test results; torque control mode

The graph above shows a typical time series from a short drive up and down a hill in torque control mode:

A. The vehicle started with full DDPM displacement and a pressure demand of 100 bar. For the acceleration phase, the DDPM was at full displacement.

B. As the vehicle accelerated, the pedal was released. As the torque demand fell, the pressure demand to the DDP fell in proportion. When the demand pressure reached the minimum pressure set-point of 20 bar, the DDPM reduced displacement below 100% to further reduce output torque.

C. At this point, the vehicle started to go down hill. The torque demand dropped to zero and the vehicle coasted down hill for a short period. The DDPM was at 0% displacement, so all cylinders were doing LP idle cycles.

D. With the vehicle still going down hill, the brake pedal was applied. The DDPM displacement went negative, meaning it was acting to pump fluid into the high-pressure line. As the actual pressure rose above the set-point, the solenoid relief was

operated by the 'bang-bang' controller. This produced a saw-tooth waveform to the actual pressure as the relief valve pulsed on and off. Because the DDPM is trying to maintain a steady output torque (product of displacement and pressure), its displacement is modulated in the inverse of the saw-tooth pattern.

E. The hill is coming to an end and the vehicle coasts for a while.

F. The brake pedal is pressed, the DDPM goes to full pumping displacement, and the vehicle stops quickly. Some portion of the kinetic energy of the vehicle has been transferred into the accumulator.

4.10.2 Speed control mode

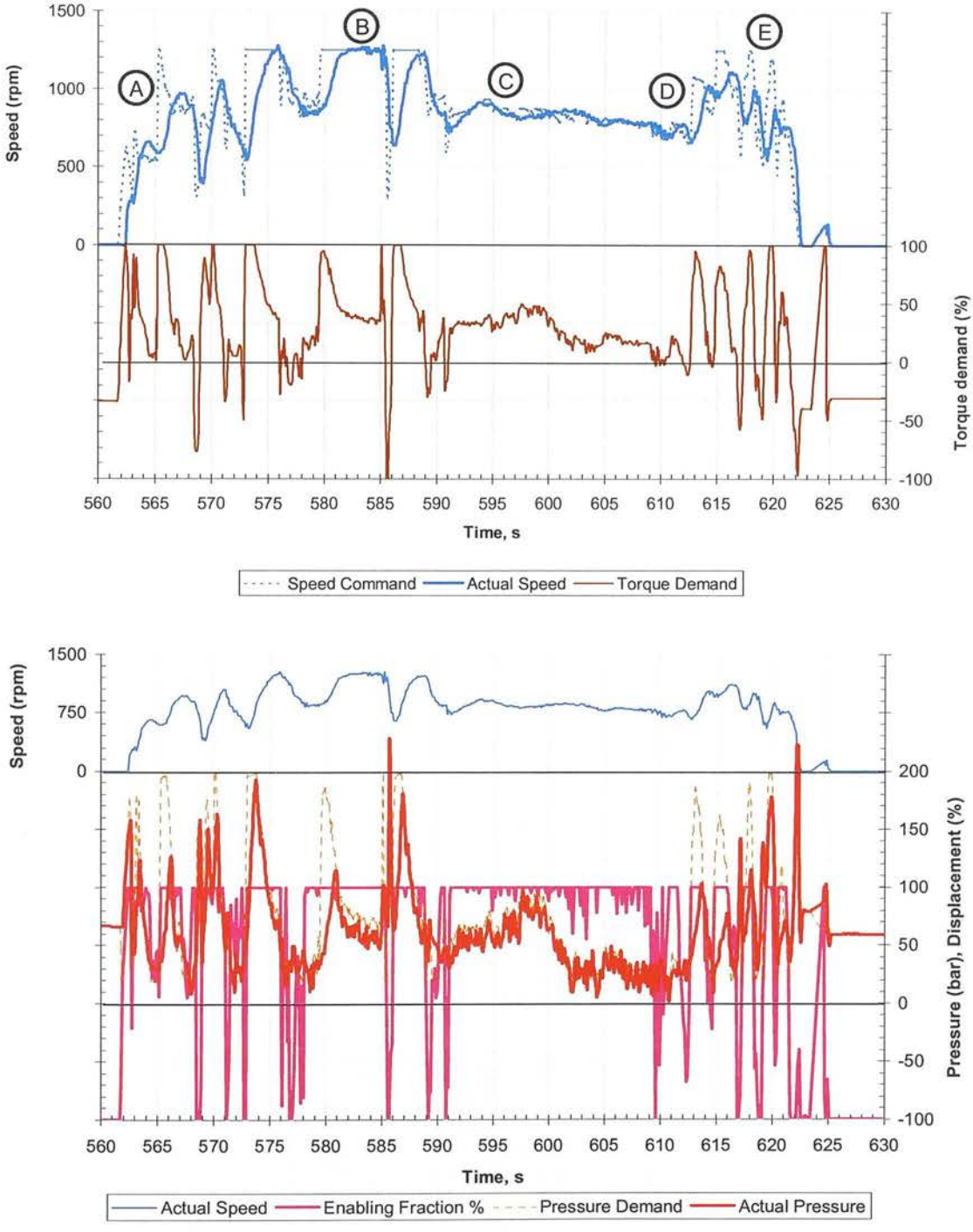


Figure 4-32: Propel DDPM vehicle test results; speed control mode

For clarity the results from the speed control tests are presented as two graphs. The top graph shows the function of the PID speed controller, the output of which is a torque demand. The bottom graph shows the function of the torque control loop, which takes the torque demand as in input and creates the DDP pressure set-point and the DDPM displacement.

Referring to the labels in the figures above:

A. The speed control pedal was given a series of steps. This produced sharp changes of the torque demand as the speed controller tried to meet the demand.

B. The response is shown to a speed demand step from 800rpm to 1300rpm.

C. A steady speed demand was given as the vehicle went over the brow of a hill. The steady-state torque required to maintain the demanded speed fell as the load on the motor reduced.

D. The vehicle was facing down hill, so the torque demand tended towards zero.

E. A series of speed demand steps were given and the vehicle attempted to maintain the demanded speed, causing alternating positive and negative torque demand.

A speed control experiment was also done in both directions, to demonstrate that the DDPM is capable of four-quadrant operation.

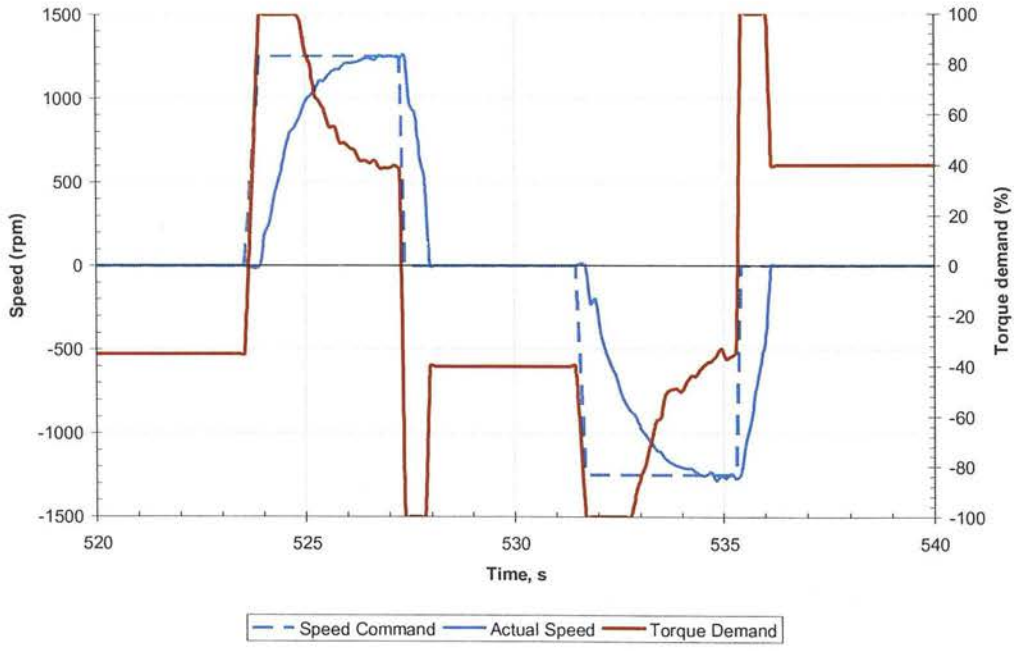


Figure 4-33 Speed control mode in both directions

The figure above shows the response of the vehicle to a step demand of vehicle speed in both the forwards (positive) direction and the reverse (negative) direction. When the positive step demand to 1250rpm is given (t=523.5s), the torque demand immediately saturated at +100% and the vehicle started to accelerate. As the vehicle approached the demanded speed, the torque demand reduced to a steady-state demand of approximately 40%. The vehicle speed

response to the demand step can be characterised as a first-order time constant of τ=0.8s; the speed controller gains were low for this test to produce this over-damped response. When the step demand to zero speed was given (t=527.2s), the torque demand immediately saturated at -100% and the vehicle decelerated to zero speed. The response of the vehicle to a negative speed demand (t=531.5s) was substantially symmetrical to the response with a positive demand.

These results demonstrate that the propel DDPM is capable of operating in all four quadrants of speed and torque.

4.11 Conclusions

The objectives of this chapter have been achieved. A DDPM has been demonstrated to be capable of propelling a vehicle. In particular:

- A method had been demonstrated which allows the DDPM to operate as a motor for propelling a vehicle. The "false rev" method exhibited some shortcomings; these are likely to be improved with further development of the embedded control software.
- The overclamped piston pad bearing is shown to be unsuitable for the propel DDPM due to its inability to combine the required features of low stiction, low leakage and stability with respect to variations in parameters. An underclamped piston pad with a capillary impedance achieved an acceptable combination of these requirements, due to the central loading inherent in the kinematic design of the radial piston machine considered.
- A vehicle propel transmission has been demonstrated comprising a pressure-controlled DDP and a secondary-controlled DDPM, with an electronic load-sensing controller. The key features required of the embedded software for the DDPM and the system controllers have been described. The propel DDPM is shown to be capable of starting from zero speed and operating in all four quadrants. Open-loop control of motor torque and closed-loop control of vehicle speed has been demonstrated.

5 The DDP propel demonstrator vehicle

5.1 Introduction

It was thought that a DDP powering a conventional motor could exhibit many of the attractive features of a complete DDPM transmission, while being less complex, and therefore closer to commercial application. Therefore, once the DDPM propel demonstration was complete, the DDPM was removed and replaced with a conventional motor. The aim of the resulting DDP propel demonstrator vehicle was to investigate the application of a DDP as a propel pump in a hydrostatic vehicle transmission, at a scale which was easy to work on.

Conventional hydrostatic propel systems fall into two broad categories, namely open circuit and closed circuit.

For propel-dominated applications such as wheel loaders and forklift trucks, closed circuit is the universal choice. Here a variable displacement pump (usually of axial piston design) is dedicated solely to propel, with working functions being supplied from an auxiliary pump. Vehicle speed is controlled by swash angle, direction being changed by this angle going over-centre such that the flow reverses at the ports of the pump. Motion control is achieved by controlling swashplate position with an hydraulic servo, often with an electrical interface. Braking is achieved by pump and motor roles reversing, some portion of the braking energy being transferred back to the prime mover, eventually to be dissipated in its cooling circuit.

Open circuit propel is more common in applications where the duty of the propel system is comparatively light, such as aerial lifts and excavators. Here the motors are typically controlled by proportional directional valves, the flow being generated by a pump which is shared between propel and working functions. Motion control comes from the proportional valve; this is typically fitted with a flow compensator such that vehicle speed can be controlled regardless of load. Typically a load-sensing arrangement is employed to share pump flow between propel and working functions; the pump may be of fixed displacement or pressure compensated variable displacement; in both cases it is the valve, not the pump, that is controlled by the operator. Braking is achieved by a valve to create backpressure on the motor, the braking energy being dissipated as heat in the oil; this may be the proportional valve itself or a separate counterbalance valve.

By contrast the DDP has unique characteristics which dictate how the machine is applied to propel:

A DDP has a P and a T port, rather than A and B ports. A directional valve arrangement is needed to change vehicle direction, unless the motor can change direction internally by for instance the swashplate moving over-centre.

A DDP can provide high-pressure flow from the P port, but cannot absorb high-pressure flow. A valve is needed to provide backpressure to the motor to brake the vehicle.

A DDP can change output flow very quickly with good open-loop linearity and zero hysteresis. It is possible to change motor displacement suddenly and compensate for this by instantaneously changing the pump displacement. Thus a switched-displacement ('two-speed') motor can be driven by the DDP without the displacement switching being detectable to the operator.

For these reasons an unusual system architecture was adopted for the demonstrator vehicle. Like typical closed-circuit systems, motion control comes from pump displacement control, so the vehicle motion closely reflects the control of pump displacement. Like a typical open circuit, braking is achieved by dissipating energy in a valve, and directional control comes from a directional valve external to the pump. As the pump has full proportional control, this valve can be two-position rather than proportional. This unusual transmission circuit could be called 'pump-controlled open circuit'.

The demonstrator was built to answer the following questions posed by this arrangement:

Could satisfactory vehicle performance be achieved by a transmission comprising a DDP, a valve circuit and a conventional axial-piston motor? At the outset it was not clear that this was achievable. The major risk was seen to be that the pump flow pulses which are inherent in the DDP principle, would feed through the driveline, causing vibrations which were detectable by the driver.

Could the DDP react fast enough to sudden displacement changes, such that the displacement of the motor can be switched while the vehicle is in motion, without creating jerk? The sudden displacement change of the motor would need to be synchronised with a sudden step in pump displacement. The presence of an overcentre valve in the circuit would mean that if the pump flow was lower than the motor flow for a significant period of time, the vehicle would decelerate sharply.

Could open-loop control of vehicle position be achieved? The DDP has control of delivered volume, rather than flowrate; this could allow position control of a vehicle.

Could the vehicle change direction quickly and smoothly? The timing of the command to the digital directional valve would have to be precise to ensure continuous acceleration through zero speed.

Could an overcentre valve be relied upon to control deceleration smoothly? Overcentre valves have the reputation of introducing instability into systems.

This chapter seeks to answer these questions.

5.2 Description of system

Figure 5-1: The converted DDP propel vehicle being demonstrated by the author to Dave Anderson, Sauer-Danfoss CEO, at the test track in Nordborg, Denmark

The donor vehicle was a Yamaha G16a golf buggy. This was chosen for the following reasons:

- Basic sizing calculations showed that, with a suitable motor, a 12cc propel DDP would be a good match for the vehicle mass and engine power.
- The vehicle was constructed with a tubular steel chassis and a removable plastic body; this made access easy for the conversion.
- The vehicle had simple mechanical steering and braking systems, so these would not have to be considered as part of the hydraulic system design.

5.2.1 Overall system design

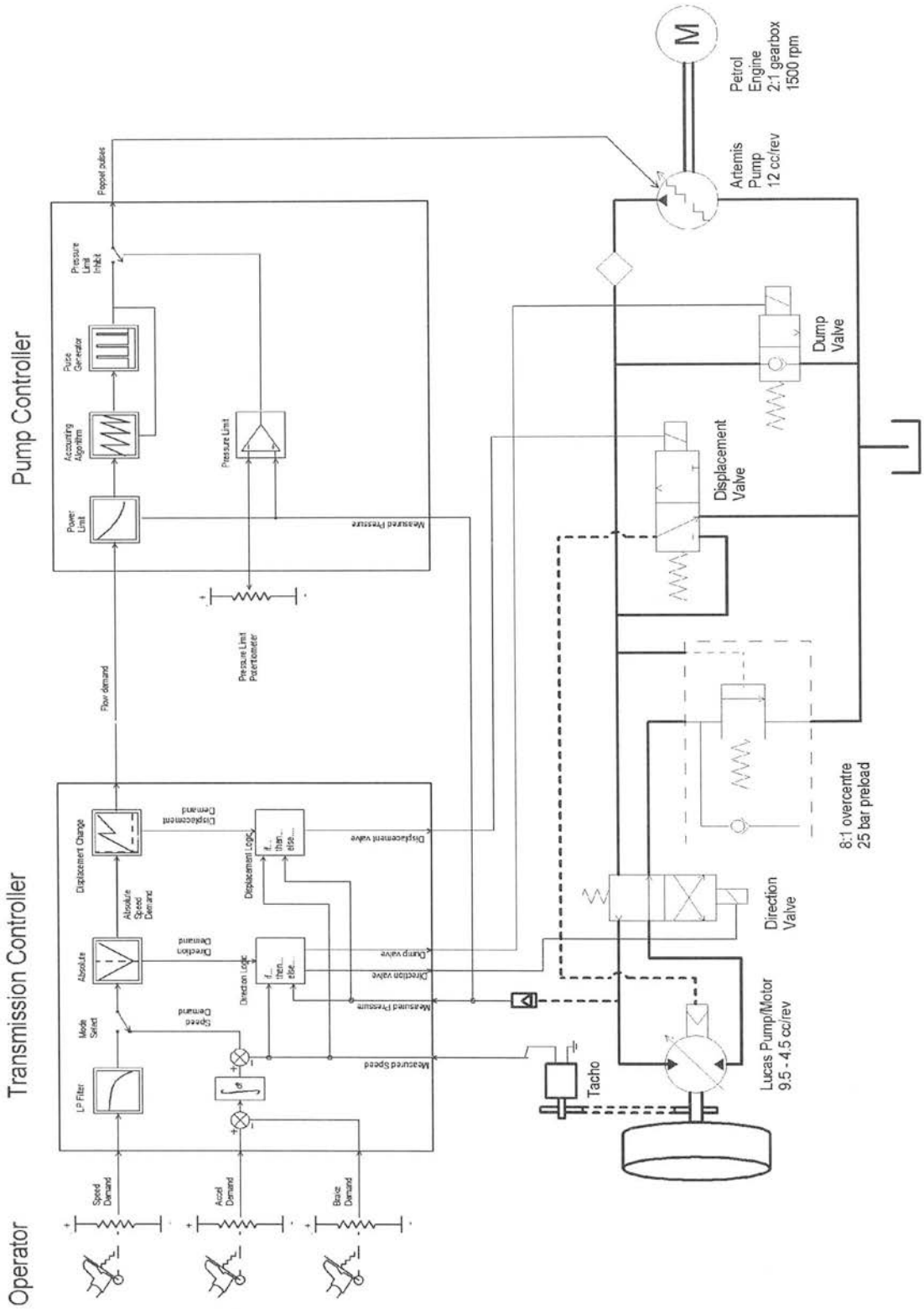


Figure 5-2: Overall system schematic

The overall system schematic is shown above. It consisted of three important sub-systems: the hydraulic circuit, the DDP controller, and the system controller (or 'transmission controller').

The Hydraulic Circuit

The pump was a 12cc/rev radial piston DDP composed of 6 cylinders each of 2cc nominal capacity, driven by a petrol engine. The case of this pump was at atmospheric pressure; a header tank of 1 litre capacity was placed 100mm above the DDP. A system relief valve limited pressure at the pump outlet to 220 bar; this is not shown in the schematic above. A high-pressure filter was fitted between the DDP and the rest of the circuit. The motor was a 9.5cc/rev axial piston unit (Lucas PM125); this was adapted to be switchable between full (9.5cc) and reduced (5.5cc) displacement according to a hydraulic pilot signal switched between pump and tank by a solenoid valve; details of this are given below. The DDP supplied the motor via a 2-position 4-port directional solenoid valve. Downstream of this valve, an 8:1 overcentre valve created back-pressure on the motor according to a pilot signal from the pump outlet; the relief pressure of this valve was set to 200 bar. Flow from the overcentre valve returned to the DDP case via a hose. The 'dump valve' shown was a remnant from the earlier DDPM transmission; it was not used and can be ignored.

The DDP Controller

The DDP controller was based on an Intel 80196 microcontroller, and designed by Jon Almond. The solenoid drivers were as for the DDPM described in Section 3.2, but flywheel resistors were not fitted because there was no need to quickly de-energise the LPV coils. The drivers were triggered directly from digital outputs of the microcontroller. A single pulse per rev from an inductive pickup sensing a pin on the crankshaft provided shaft position feedback to the controller.

The DDP controller varied the displacement of the DDP according to a voltage signal from the transmission controller; details of this interface are given below. A software pressure limiter in the controller reduced the displacement of the DDP above 200 bar as sensed by a pressure transducer.

The System Controller

A separate Intel 80196 controller was in overall charge of the vehicle behaviour. This read operator commands and synthesised a displacement demand signal for the DDP controller, with feedback of vehicle speed and pressure. It also actuated the directional and displacement change solenoid valves; this is described in detail below.

Table of important vehicle parameters

Pump Capacity	12 cc/rev
Motor capacity, full displacement	9.5cc/rev
Motor capacity, reduced displacement	5.5 cc/rev
Gear ratio, engine: pump	2:1
Engine maximum speed governor setting	3800rpm
Maximum engine power @ 3000rpm	7.6kW
Maximum 2:1 gearbox output torque (=maximum permissible pump torque)	48.4 Nm
Pump software pressure limit setting	200 bar
Maximum pump torque (200 bar, full displacement)	38Nm
Pump pressure relief valve setting	220 bar
Rolling radius of wheel	199mm
Differential ratio	10:1
Timing belt drive ratio - motor: differential input	40:48
Mass of vehicle plus driver	450kg

Figure 5-3: Table of important parameters

5.2.2 Mechanical design

A solid model of the rear portion of the buggy was created in Solidworks 99, allowing the mounting of the engine, pump and motor to be designed in the context of the constraints of the existing vehicle structure.

Figure 5-4: Solid model of the rear section of the golf buggy showing overall layout

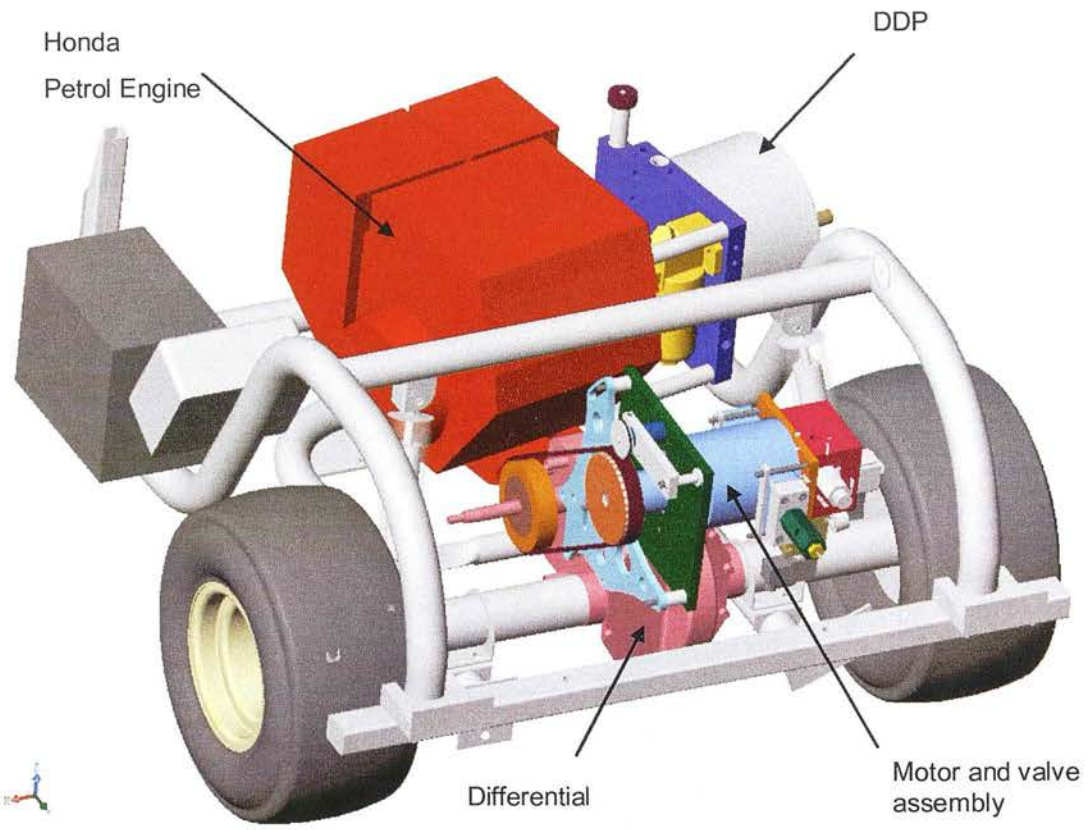


Figure 5-5: Major system components

The original Yamaha engine was not suitable as a prime mover as it lacked both a step-down gearbox (the nominal DDP speed was 1500rpm) and a speed governor. It was replaced with a Honda GX340, a single-cylinder four-stroke petrol engine with a 2:1 step-down gearbox, speed governor and electric starter.

The DDP was mounted to the engine with a plate and spacers; drive from the output shaft of the engine gearbox was via a rubber coupling. The engine/pump assembly was mounted to the chassis at an angle to accommodate the hoses and electrical connections at the end of the DDP. The mechanical mounting of the pump to the engine was designed by Carn Gibson.

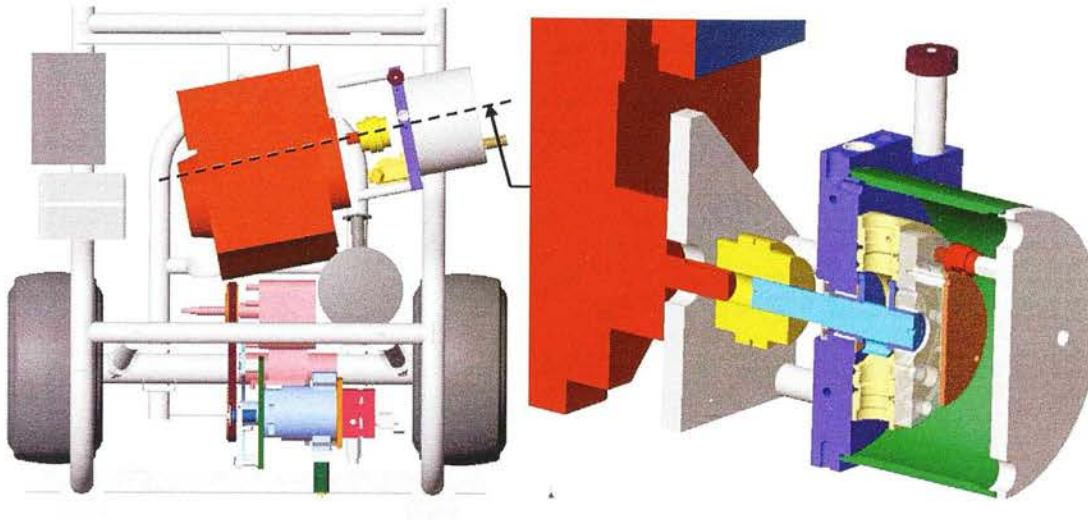


Figure 5-6: Plan view of buggy and section view of DDP mounted on engine

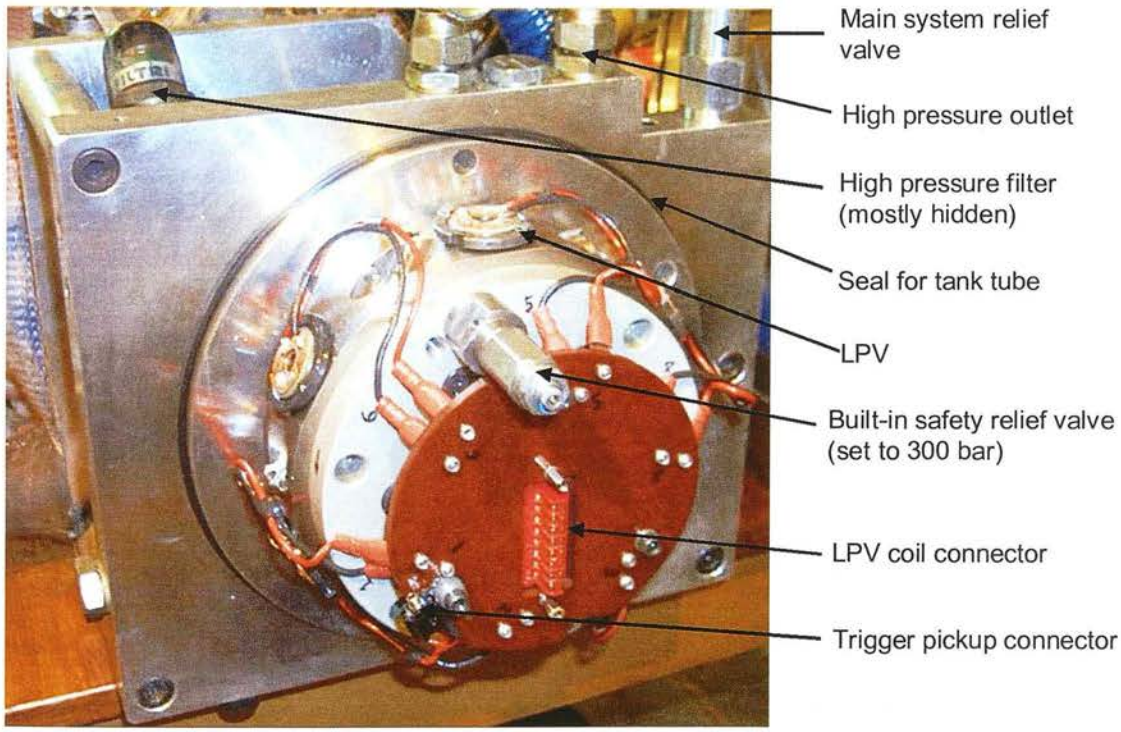


Figure 5-7: The golf buggy DDP, with tank tube removed

Because the original transmission was a belt continuously variable transmission (CVT), the input shaft to the differential was parallel to the axle. The easiest way to drive this with a hydraulic motor was seen to be with a synchronous belt drive, placing the shaft of the motor parallel with the axle. This also allowed another gearing ratio to be built into the driveline by choosing the number of teeth on the motor and differential pulleys, allowing overall drive ratio between motor shaft and axle to be chosen. A laser-cut steel plate located the motor onto hard points on the differential case. A slot in the plate allowed the distance between motor

shaft and differential shafts to be adjusted so that the belt could be properly tensioned. A tachogenerator was driven by a wheel resting on the drive belt at the motor pulley to measure the vehicle speed; this was the main vehicle speed sensor for both embedded control and data acquisition purposes. The valve block was mounted to the motor and connected to the DDP with flexible hoses.

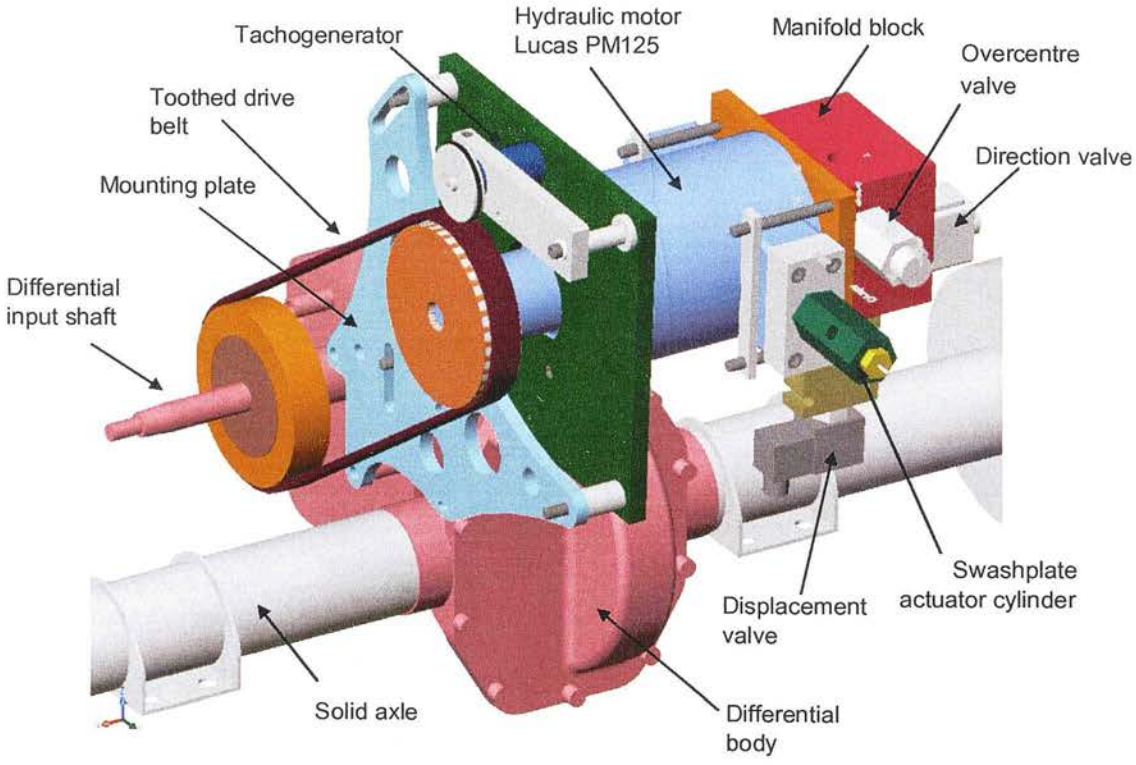


Figure 5-8: Design of the mounting of the Lucas motor

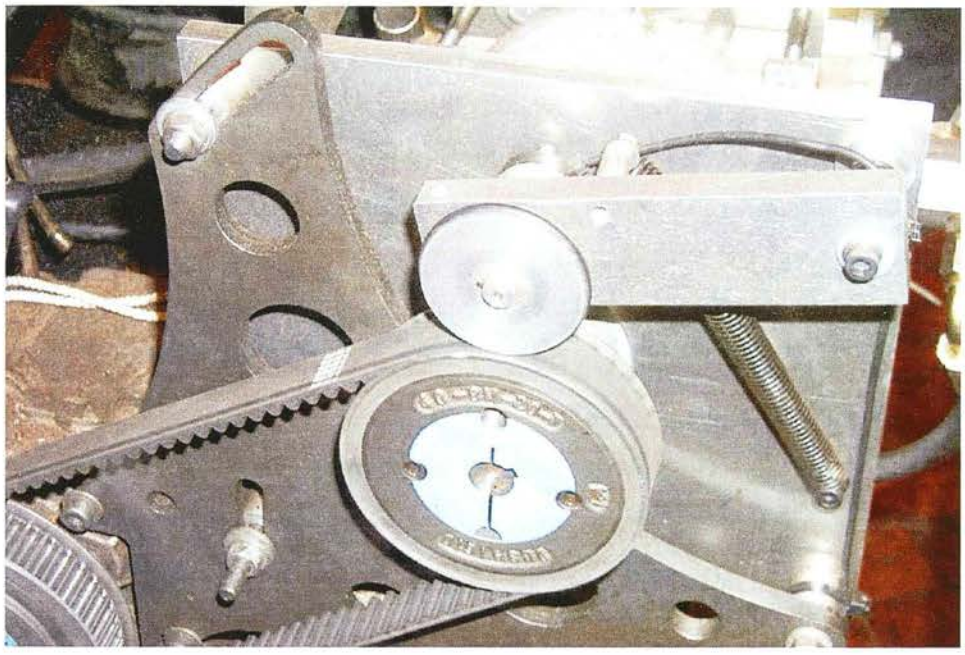


Figure 5-9: Close-up of the belt drive and tachogenerator arrangement

The propel motor was a Lucas PM125 variable-displacement axial piston swashplate unit of maximum displacement 9.5cc/rev. A servo piston moved the swashplate to the position demanded by the servo rod. Swashplate actuation pressure came from the main pump pressure port.

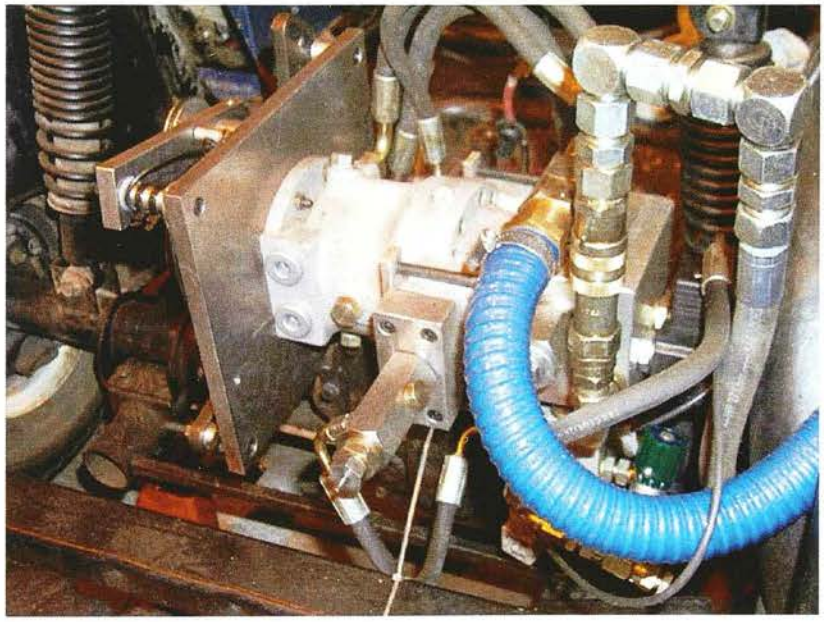


Figure 5-10: View from the back of the buggy showing displacement actuator and manifold block

In order to switch the motor between two positions, a servo actuator cylinder was made to replace the original handwheel adjustment of the servo rod position. When the displacement

control solenoid valve was energised, this cylinder was connected to fluid pressure, creating an inwards force which pushed the servo rod to its endstop, causing the main swashplate actuator piston to push the swashplate towards the neutral position. An end-stop was fitted to the main swashplate actuator piston to limit the minimum displacement to 5.5cc/rev. When the displacement control valve was not energised, the servo actuator cylinder was connected to tank, and the servo rod pushed to its outer end-stop, causing the motor to switch to full displacement. An indicator pin allowed the displacement actuation to be observed.

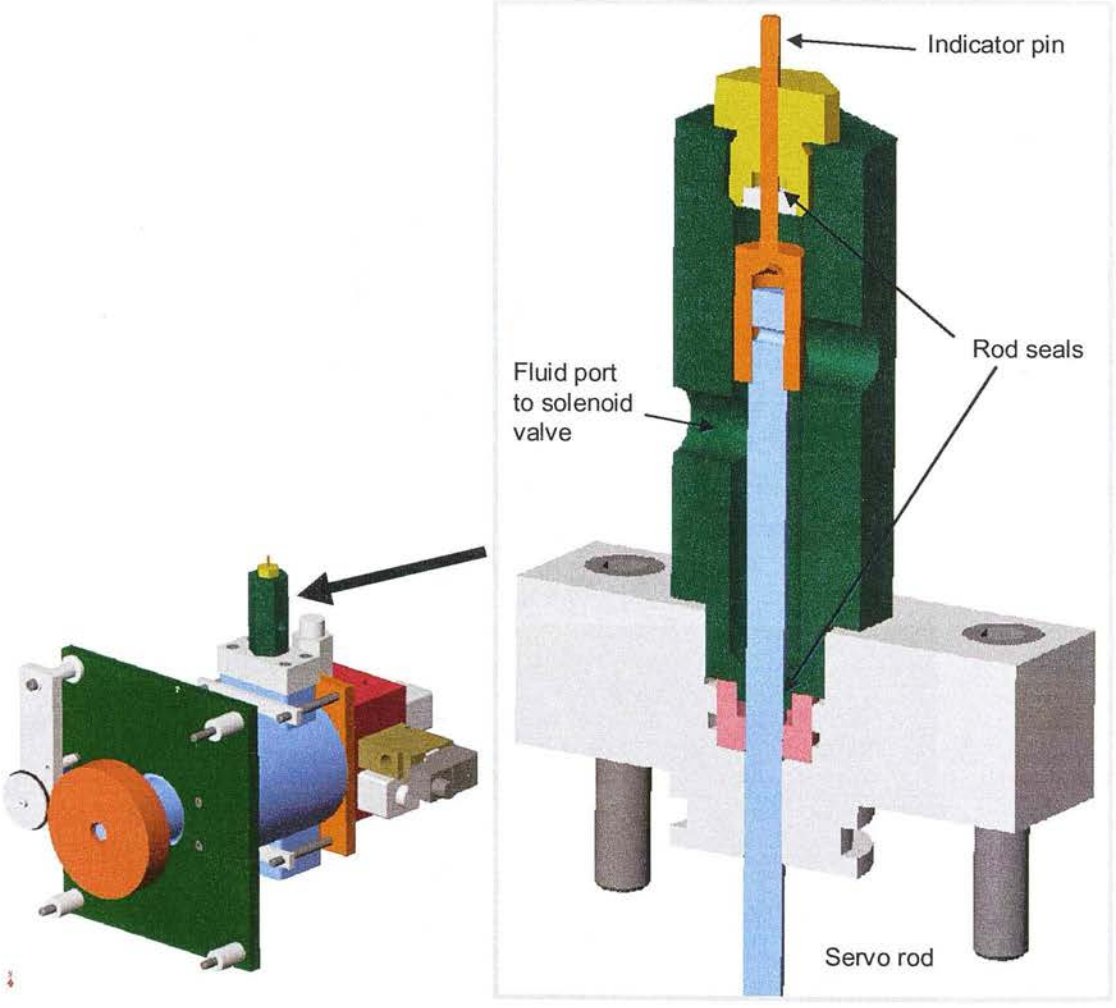


Figure 5-11: Two-speed displacement actuator made for the Lucas motor

5.3 Interface and control design

5.3.1 DDP Flow Algorithm

In the original DDP flow control algorithm, as described by Rampen (PhD thesis 1992, p.16), the controller had one choice every time a cylinder approached bottom dead centre i.e. to pump, or not to pump. If the accumulated error between demanded volume and delivered volume exceeded half of a cylinder volume, then the next available LPV would be closed at BDC and a half-sinusoid of flow would be delivered; otherwise that LPV was left open and that contracting stroke instead delivered fluid back into the low-pressure tank.

It is obvious that this mixing of full and idle strokes becomes increasingly pulsatile as the displacement of the machine approaches zero, and the ratio of pumping cylinders to idle cylinders approaches zero. Time(s)

Figure 5-12 below (top graph p. 243) shows the simulated flow output of a 6 cylinder DDP running at 1500rpm following this algorithm (this was generated by a model implemented in Dymola, described in Section 6).

The significance of this behaviour depends on the use to which the pump is put. For instance, an electrically-driven industrial power-pack may have the task of maintaining a pressure line at 200 bar, to supply a number of servo valves controlling the cylinders of a test stand. Here it is possible fit a large gas-filled accumulator to the output of the pump, acting to reduce the pressure ripple caused by the discontinuous flow of the DDP. The combination of the orifice characteristic of the spool in the servo valve and the large compliance of the accumulator form a low-pass filter. Hence it is likely that what pressure ripple remains at the pump outlet, smoothed by the accumulator compliance, will be further reduced in the cylinder chamber, so that the pulsatile character of the DDP flow will not cause detectable vibration on the mechanism actuated by the cylinder.

Vehicle propel is however fundamentally different. In vehicle such as a wheel loader, fine control of vehicle position is required with a high response speed. Hence such systems are usually designed to be hydraulically stiff, while the load presented to the motor is predominantly inertial, so the flow taken by the motor from the system compliance changes slowly. The difference between the pulsatile DDP flow and the slowly changing motor flow must flow into or from the system compliance. Due to the high stiffness of the connection between pump and motor, this translates into a large pressure pulsation, causing large propel torque pulsations. The frequency of these pulsations increases as the flow increases. At low vehicle speed the flow comes in discrete pulses, each of may be felt by the driver if they are

of sufficient magnitude to be noticeable above the normal vibration levels on such machines caused by, say a diesel engine.

An obvious solution to this problem is to increase the frequency and decrease the magnitude of the pulses. A DDP can be fitted with a large number of small cylinders, but this adds significantly to the complexity and cost. It is desirable to limit the number of cylinders to 7 or less so that they can be packaged into a single bank radial piston arrangement. The prototype DDP machines had 6 cylinder banks and it was extremely important to establish whether this was sufficient for a propel service.

Another possible solution is to fit the DDP with a number of cylinders of different sizes, and use the smaller cylinders at low displacement demands. However this again increases complexity and cost.

There is one aspect of the machine is easy to change and costs nothing to implement-changing the embedded control algorithm. Two alternative algorithms were considered as an improvement on the pulsation behaviour of the basic algorithm, namely:

1. **Variable part stroke.** It is possible to enable every cylinder on every stroke, and control the displacement of the pump by changing the timing of the LPV actuation. Time(s)

Figure 5-12 below (bottom graph p. 243) shows the simulated flow output, and Time(s)

Figure 5-13 (bottom graph p. 244) shows a zoomed-in view for the first 20% of displacement.

2. **Fixed part stroke.** A fixed "part stroke" fraction is established and these stokes are mixed with idle strokes at low displacement demands. When the displacement demand exceeds the part stroke fraction, there is a transition to the full stroke algorithm (mixing full strokes and idle). Time(s)

Figure 5-12 below (middle graph p. 243) shows the simulated flow output with a fixed part stroke fraction of 17%, and Time(s)

Figure 5-13 (top graph p. 244) shows a zoomed-in view for the first 20% of displacement demand showing the transition and contrasting this with the basic full stroke enabling algorithm.

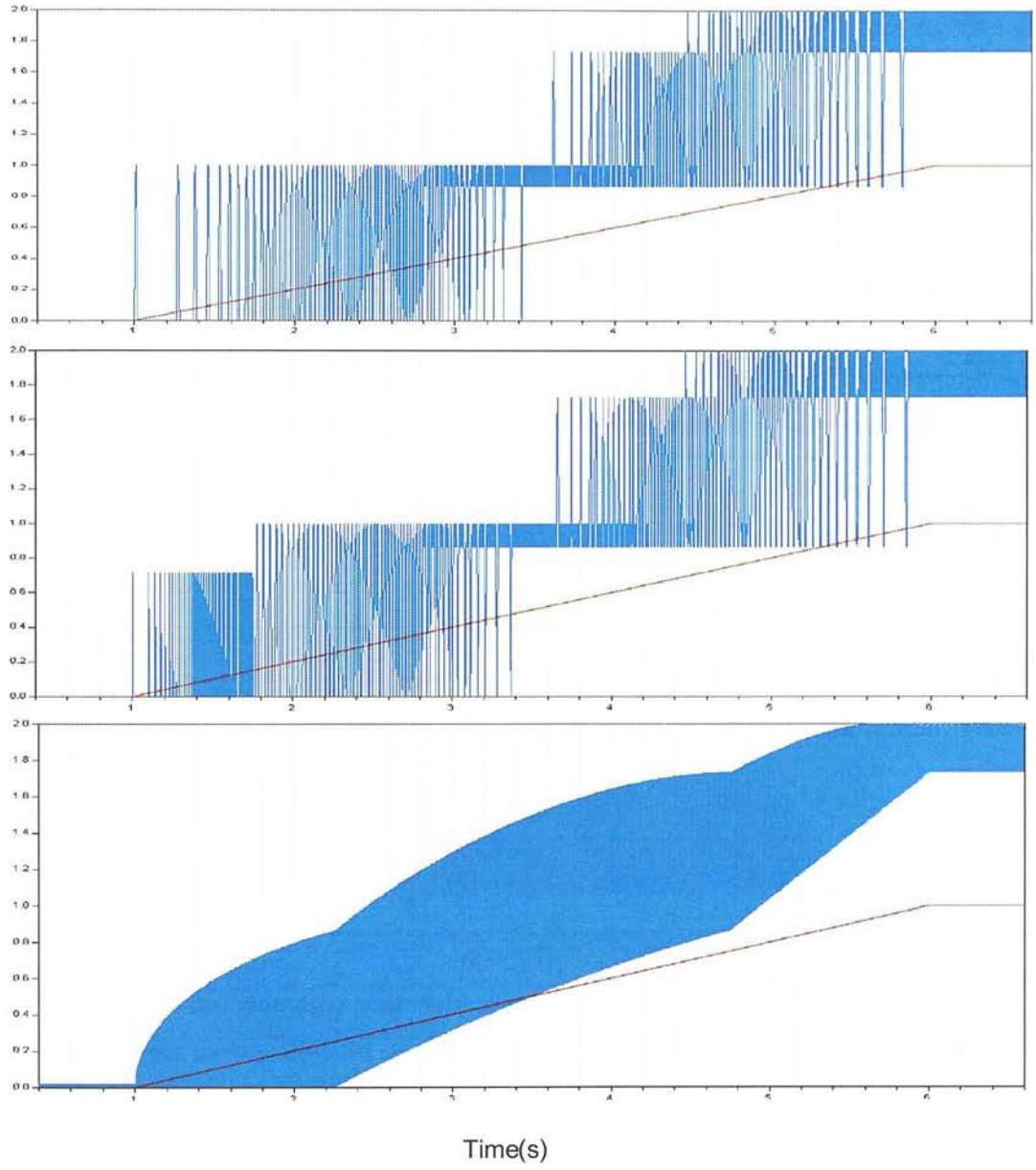


Figure 5-12: Simulation of flow output (normalised such that the peak flow from single full stroke = 1) of a 6 cylinder DDP; blue=flow output, red=displacement fraction demand; top=full strokes; middle=fixed part stroke (no mixing); bottom=variable part stroke

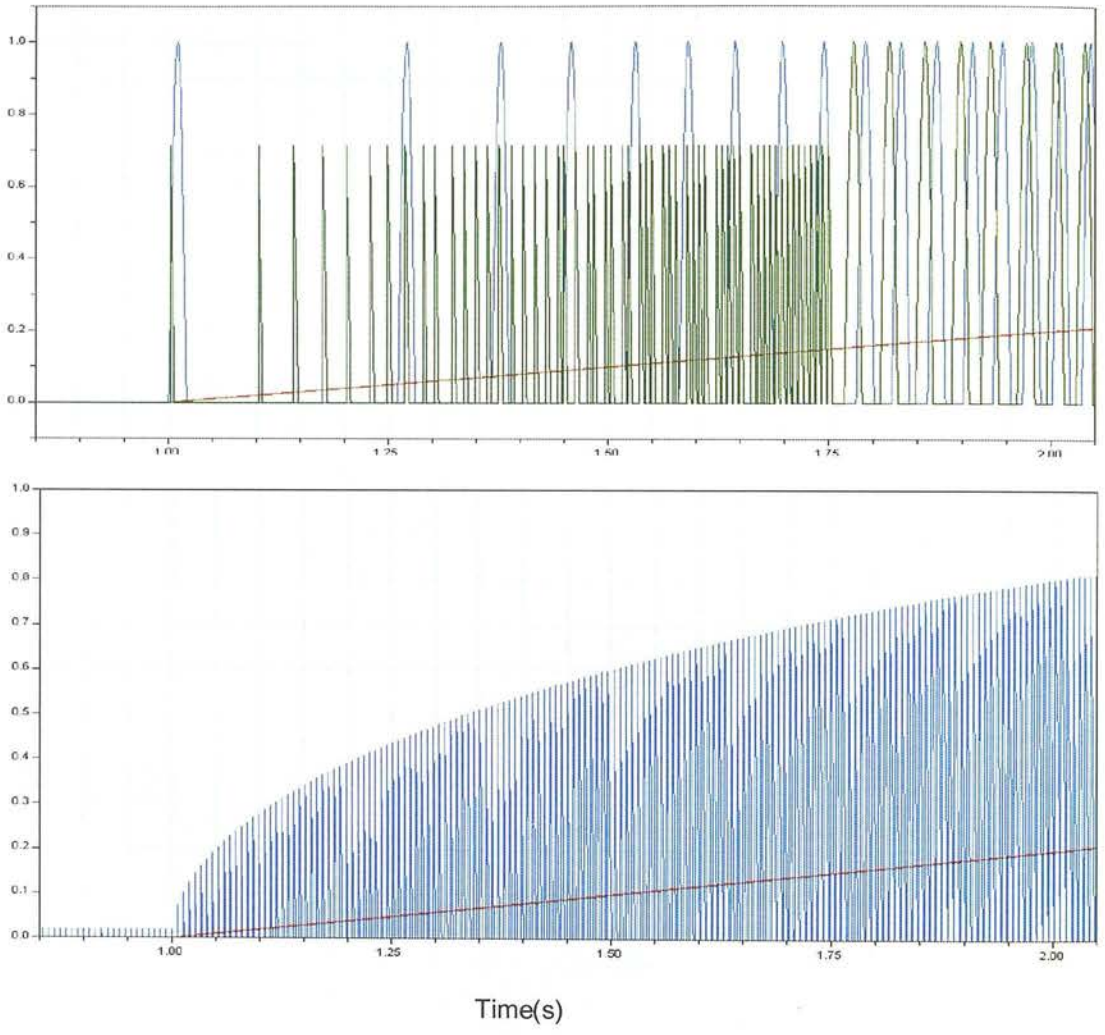


Figure 5-13: Detail of the low-flow region; top: blue=flow with only full strokes, green=flow with fixed part stroke (no mixing); bottom: blue=flow from variable part stroke

Variable part stroke seems to offer the best possible pulsation pattern from the machine, in terms of minimising the ratio between the peak and mean flowrate across the full displacement range. Dynex Ltd. produce a check-valve commutated piston pump for industrial use, in which displacement is adjusted by continuous timing variation; this is achieved purely mechanically by a cam arrangement. However for a DDP, this method suffers from some important disadvantages.

At very small displacement demands, the LPV would be closing a very short time before TDC. While there was no worry that the timing of the pulses to the valves could be controlled consistently, the valves themselves had a finite transit time and this could never be totally repeatable. Not only were there manufacturing tolerances between individual valves, but the transit time could change with temperature due to the effect of fluid damping and flow forces

on the valve motion. As the flow reduced towards zero, this would mean increasing uncertainty about the exact amount of fluid displaced to the load.

This timing uncertainty would be further complicated by the effect of compressibility of the dead volume in the cylinder, which was around 2.8 times the swept volume. Given that oil must be compressed 1.25% to raise the pressure to 200 bar, this would have a dominant effect at very low displacement fractions. If the timing of the LPV closure was set to give 1% displacement at 10 bar, the pump would deliver no oil at 200 bar because the pressure in the chamber would never reach as high as the fluid upstream of the HPV; leakage would exacerbate this effect. To get consistent behaviour at all pressures, the pump controller would have to actively adjust the LPV timing as a function of pressure.

With very small part strokes, the LPV closes just before to TDC, and the flow is generated with lower piston velocity than is average throughout a full sinusoidal flow pulse. This increases the ratio of amount of time during which the cylinder is pressurised, to the total flow from the pump. The increase of this ratio means that the effect of leakage from the cylinder becomes greater as the part stroke fraction decreases; it is likely that this would reduce efficiency.

For the variable part stroke mode, the LPV must be able to close at any point during the pumping stroke, including the time of maximum fluid velocity, halfway through the stroke, causing a substantial shock to the valves and the structure of the pump. Experience with the Dynex pump showed that this would be a noisy mode of operation. Mechanical considerations may be even more important. It was fundamental to the construction of the LPVs used in the DDPs made at this time, that a relatively heavy ferromagnetic armature had to be attached to a light-weight polymer poppet. When the poppet hit the seat, the armature has to be decelerated suddenly, causing stress at the attachment between poppet (which suddenly stops moving) and the armature. Failure of this interface had been observed in operation of the prototype DDPM (see Appendix 9.4) and it was thought that this method of displacement control would drastically shorten the lifetime of the LPVs.

Ultimately the risk of damaging one of the few DDPs in existence ruled out much experimental work on the variable part stroke algorithm, but it is deserving of further study and is investigated by simulation in Section 6. Experimental attention focussed instead on the fixed part stroke algorithm.

The choice of the fixed part stroke magnitude was in some ways arbitrary but experiments indicated that noise increased quickly above 20%, while fractions under 10% yielded less than consistent results. It was realised that a natural fraction presents itself when trying to implement code which allows the transition from part stroke mode to full stroke mode. In a 6

cylinder DDP running at constant speed, there are 6 possible events per shaft revolution. These events are timed by setting an interrupt-driven timer, phase locked to a once-per-rev pulse from a sensor on the shaft (the "shaft trigger"). In full stroke mode, if the cylinders are numbered 1 through 6, then the first LPV to be energised after the shaft trigger is #1. However if at this time a pulse is sent instead to cylinder 5 (which is phase shifted by -120°) then that cylinder will produce a partial stroke of around 17%. This fraction was chosen for all subsequent experimental work for convenience, to reduce the DDP embedded software complexity.

Although the demonstrator vehicle engine was governed to a nominal speed of 1500rpm, shaft speed variations were expected as a function of engine load, so it was important as a first step that an algorithm was implement to vary the timing of the LPV pulses as a function of shaft speed, so that the LPV closed at exactly the desired phase angle.

An experiment was carried out to find the timing necessary to maintain a 17% part stroke. The DDP was installed in the vehicle, driven by the Honda engine and delivering flow to the axial piston motor, the speed of which was measured with the tachogenerator. At each engine speed, the timing was adjusted to obtain the maximum motor speed with all cylinders pumping full strokes. The software was then changed so that all cylinders pumped part strokes, and the timing was adjusted to obtain exactly 17% of the motor speed at full flow.

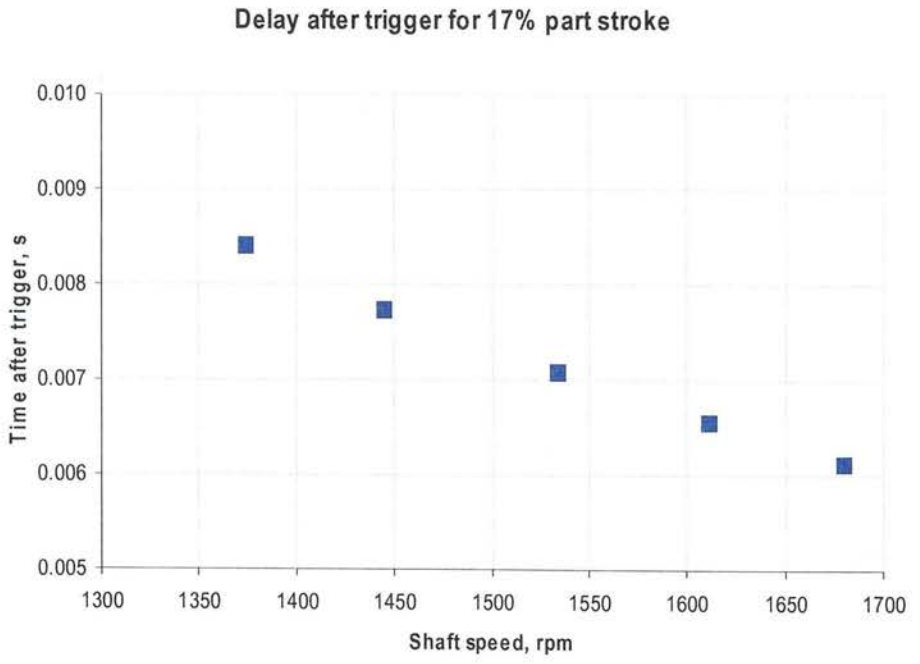


Figure 5-14: Calibration of the part stroke timing

The graph above shows the time delay after the shaft trigger (aligned with BDC of cylinder 6) at which the LPV had to be fired, to achieve 17% part strokes; the timing offset was reasonable linear with speed. A simple calculation was added to the DDP controller to implement linear timing variation with respect to speed, effectively calibrating the DDP so that it would deliver 17% part stroke over the full range of shaft speeds.

The effective delay between the start of the pulse to the LPV and the poppet hitting its seat was not known. However from the data collected it is possible to estimate this, based on the assumption that to obtain the same fraction of full flow, the valve must close at the same shaft angle. By plotting the results in terms of estimated angle of poppet impact, it was be possible to find the effective delay:

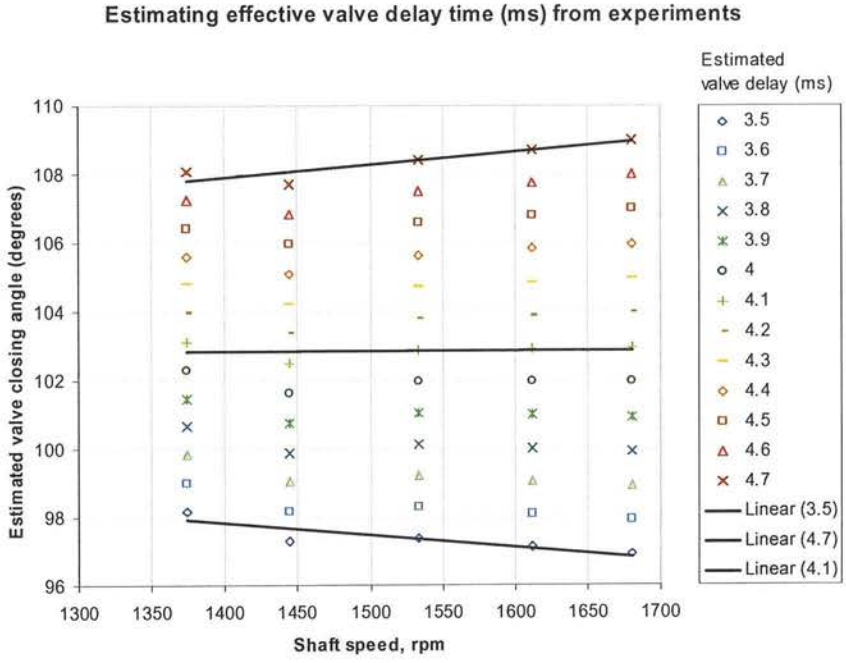


Figure 5-15: Derivation of the LPV closing time delay

It can be seen that an estimate of 4.1ms effective delay between the start of the pulse and the actual poppet impact, yields a constant angle after trigger for the poppet impact.

Once the part stroke fraction was consistent at all speeds, an experiment was carried out to find the linearity of the pump flow algorithm.

Linearity of the fixed part-stroke flow algorithm

Axial-piston pumps typically suffer from non-linearities in their response to flow demands (Sauer-Danfoss MCV105C datasheet, 1999). Modulation of the flow is typically achieved by

moving the swashplate mechanism with a hydraulic cylinder, controlled by an electro-hydraulic feedback control system. Deadzone and hysteresis are well-documented effects and can cause problems in vehicle control at low speed.

For the DDP, there is no swash control mechanism so there should, in theory, be zero deadzone or hysteresis. However the fixed part stroke algorithm relies upon an accurate calibration of the part stroke fraction, to ensure that there is no jump in flow at the transition from part stroke to full stroke mode. The importance of this is illustrated below:

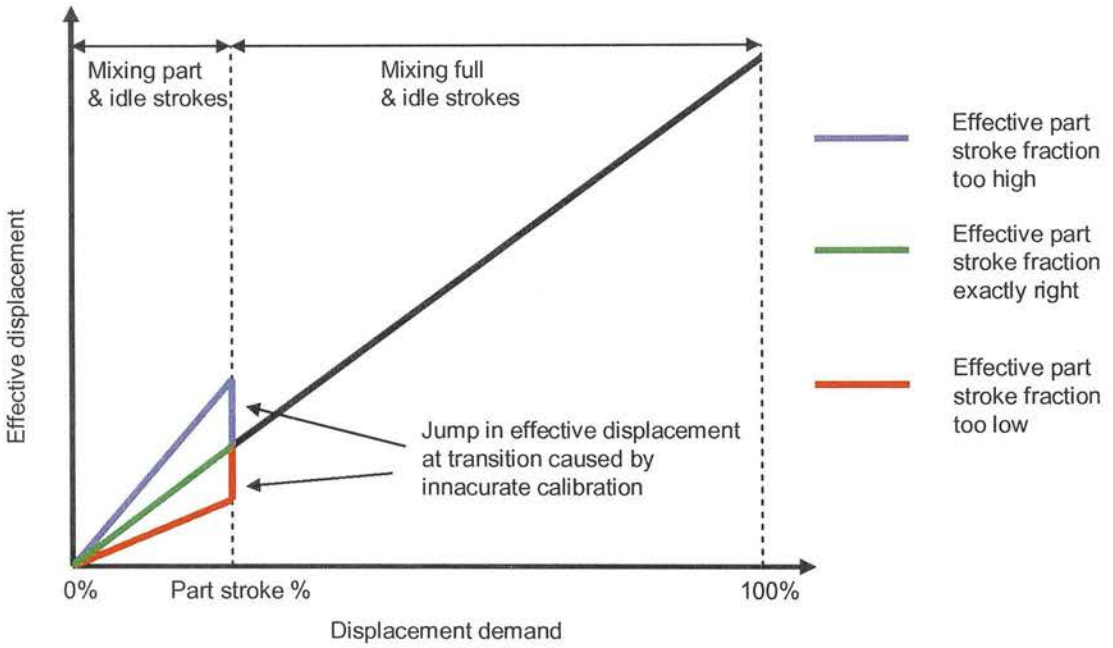


Figure 5-16: Effect of poor calibration on the flow linearity of a DDP with the fixed part stroke algorithm

Once the LPV timing was calibrated to achieve 17% part strokes, an experiment was carried out to investigate the linearity of the flow response of the DDP. The back of the vehicle was raised so that the wheels were unloaded and free to rotate. The pump command was increased in steps zero to maximum, then decreased back to zero. The motor speed was measured using the voltage from the tachogenerator. It was assumed that motor speed is proportional to pump flow, because the motor was unloaded and therefore the volumetric losses of compressibility and leakage should have been minimal.

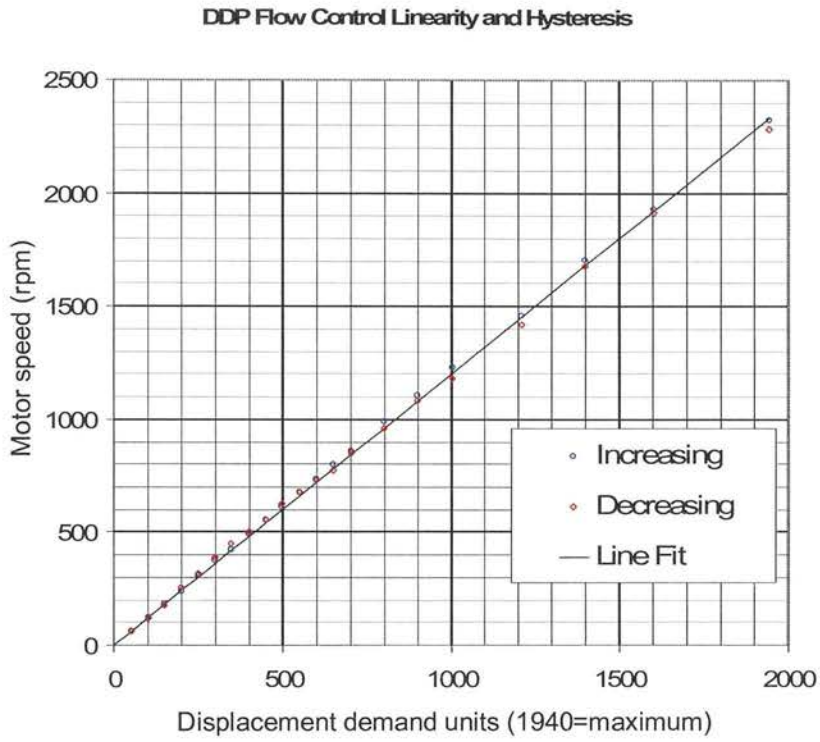


Figure 5-17: Linearity of the (unloaded) motor speed to the DDP displacement demand

The results show that the DDP does not exhibit measurable deadzone, hysteresis or non-linearity in its response to flow demand signals, at low pressure. The transition to part strokes (at a command of 330 above) did not cause a discontinuity, if the timing calibration was correct. However, at pressure other than that for which the part stroke fraction is calibrated, the effect of fluid compressibility can introduce a non-linearity at the transition point; this effect is simulated in Section 6.5.2.

5.3.2 Interfacing of DDP and system controller

Displacement demand signal to DDP controller

The system controller ran through its control calculations every 10ms. The pump controller was synchronised to the pump shaft, and ran through its control calculations 6 times per revolution. Therefore the two controllers were fundamentally asynchronous in operation. Both controllers possessed a single RS232 serial port, which it was desired to keep reserved for data logging and diagnostic communication with the host PC. The use of an analogue interface between the transmission and the pump controller allowed simple, direct communication which was easily diagnosed and did not require the two controllers to be synchronised.

A drawback of this approach however was the limitation of the Intel 80196 analogue output, which is only of 8-bit resolution. Given that this output had to span the full range, this would have given only 42 steps of part-stroke enabling fraction, and experiments showed noticeable quantisation effects at low speed. The solution was to define an interface in which the command voltage was proportional to the square root of flow. Thus at low flow demand there was higher accuracy than at high flow demand. This was seen to be particularly valuable at very low displacement.

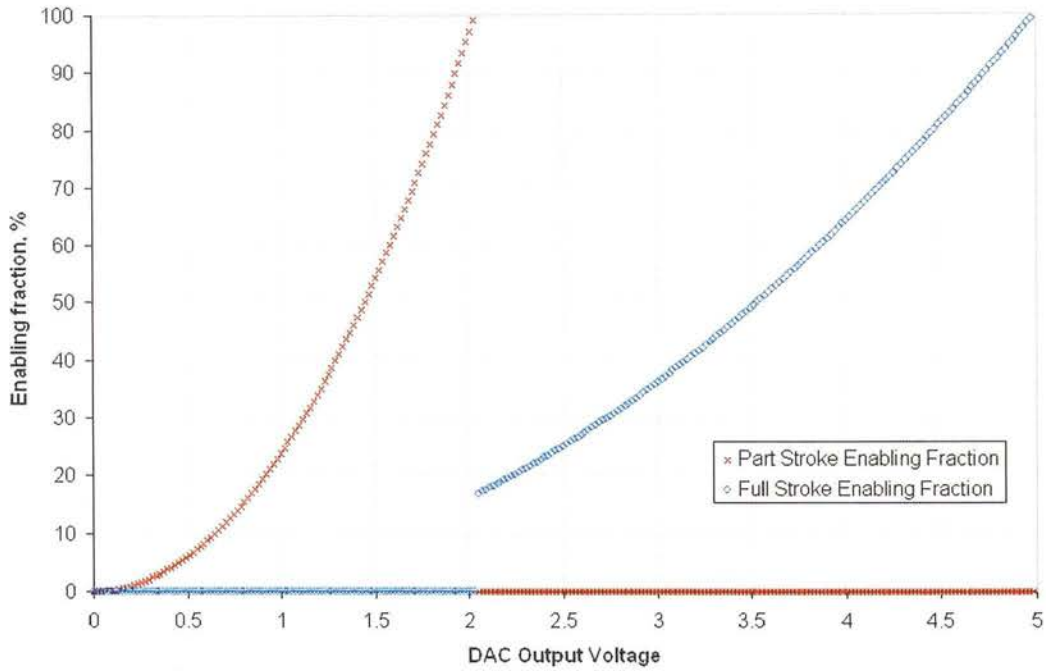


Figure 5-18: Displacement demand analogue interface; each point represents one step of the 8-bit DAC.

5.3.3 Data collection and real-time parameter adjustment

Data from the system controller was transmitted to the host PC by an RS232 serial link, running at 115 kbaud (the limit of the '196'). A library of functions was written for the '196' controller to create a software FIFO buffer. Program functions could add data to this buffer at any time, but data would not be transmitted until the calculation loop for that time step was finished. If the buffer was non-empty, then up to 10 characters would be transmitted and removed from the buffer in this period. This limited the transmitted data to 1000 characters per second, but guaranteed that data transmission did not interfere with the time-critical control functions.

At an early stage it was realised that a simple interface was needed to allow adjustment of internal system controller parameters at run-time, while the vehicle was operating. A 10-

position multi-switch selected one of ten integers between 0 and 1024, each of which was linked to an important control parameter. A potentiometer allowed any one of these to be adjusted in real time, starting from the values compiled into the C code.

When the selector was in the extreme maximum, one of these parameters controlled overall controller behaviour, placing the controller into one of three global modes of operation:

- Parameter adjustment (default). This allowed the user to adjust parameters in real time, while feedback was sent over RS232 to the PC terminal showing which parameter was being adjusted and its current value. When a new parameter was selected, the pot had to be turned until it corresponded with the current value of that parameter before the pot "latched" and took control of that parameter. This prevented parameters jumping to unintentional values when turning the selector. The allocations of these parameters to internal controller variables was chosen at compilation depending on which ones were of greatest interest at that point in the development.

```
From left to right:

Variable name          Function                                        Range, units
Tacho_deadband*10      Deadband to detect zero speed
Accel_speed_feedback   Speed feedback to accel offset
low_to_high_delay*10   Delay compensation shifting to high disp       ms
high_to_low_delay*10   Delay compensation shfting to low disp        ms
speed_pedal_range*10
Leak_time*10           Time before jerk start initated                 ms
pulse_time*10          Duration of jerk start pulse                    ms
pulse_flow             Flow demand to DDP during jerk start            0 to 1000 =
                                                                       0 to full flow
leak_speed/100         Minimum speed demand to initiate jerk start     0 to 1000 =
                                                                       0 to 10%
data_mode, input_mode  Select global controller mode

Typical output of parameter adjustment mode: (note that ">" indicates which parameter
is selected; "*" indicates that the parameter is latched and is now controlled by the
pot)

 0200  0200  0106  0108 >0200  0600  0100  0500  0400  0400          0020  S0145 P0000
 0200  0200  0106  0108  0200  0600 *0109  0500  0400  0400          0020  S0145 P0000
 0200  0200  0106  0108  0200  0600  0109 >0500  0400  0400          0020  S0145 P0000
 0200  0200  0106  0108  0200  0600  0109  0500 >0400  0400          0020  S0145 P0000
 0200  0200  0106  0108  0200  0600  0109 >0500  0400  0400          0020  S0145 P0000
 0200  0200  0106  0108  0200  0600  0109  0500 >0400  0400          0020  S0145 P0000
 0200  0200  0106  0108  0200  0600  0109  0500  0400 >0400          0020  S0145 P0000
 0200  0200  0106  0108  0200  0600  0109  0500  0400 >0400          0020  S0145 P0000
 0200  0200  0106  0108  0200  0600  0109  0500  0400 >0400          0020  S0145 P0000
 0200  0200  0106  0108  0200  0600  0109  0500  0400 >0400          0020  S0145 P0000
 0200  0200  0106  0108  0200  0600  0109  0500  0400 *0404          0020  S0145 P0000
 0200  0200  0106  0108  0200  0600  0109  0500  0400 *0473          0020  S0145 P0000
```

Figure 5-19: Run-time adjustable parameters in the system controller and an extract of the terminal log showing parameter 10 being adjusted

- Data acquisition. This mode was designed for acquiring, to the host PC terminal, regular samples of the important internal controller variables. These could then be transferred to a spreadsheet for analysis. It was still possible to adjust parameters in

this mode but there was no feedback about which parameters was being adjusted or the value of the parameter. The update rate was 11Hz.

From left to right:

Variable name	Function	Range, units
timestamp	Time since data mode started	Tenths of a second
tacho	Tacho voltage	0 to 1000 = 0 to 5V
speed_demand	Overall speed demand	-1000 to 1000
pedal_dem	Joystick signal	0 to 1024
accel_pedal	Accel Pedal signal	0 to 1024
brake_pedal	Brake Pedal signal	0 to 1024
flow_demand/1000	Flow demand to DDP	0 to 1000 = 0 to 5V (square root function)
accel_demand	Driver acceleration demand	-1024 to 1024 = -/+ 1 second ramp
Pressure_in	Pressure transducer	0-250 = 0-250bar
Shaft_posn	Position control handhweel angle	0-1024 = 0-360degrees
Filter_movement/10	Filtered derivative of above	0-1024 = 0-10 revs/s
Position_err/100	Open-loop position error	0-1024
control_mode	Current control mode	a=acceleration, s=speed, p=position, i=idle
dir_valve	Direction valve digital output	0=forwards
disp_valve	Displacement valve digital off	0=valve off (full displacement)
dump_valve	Dump valve digital output	0=valve off
tacho_direction	Polarity of tacho signal	1=forwards

Excerpt of typical data collected:

```
0110,0000,0000,0507,0106,0164,0000,-0094,0000,0189,0000,0000,a,0,0,0,1
0111,0000,0000,0507,0157,0164,0000,-0043,0000,0189,0000,0000,a,0,0,0,1
0112,0000,0000,0507,0196,0164,0000,-0004,0000,0190,0000,0000,a,0,0,0,1
0113,0000,0000,0508,0212,0164,0001,0024,0000,0188,0000,0000,a,0,0,0,1
0114,0002,0005,0508,0272,0165,0010,0143,0001,0190,0000,0000,a,0,0,0,1
0115,0000,0016,0507,0329,0164,0029,0258,0000,0189,0000,0000,a,0,0,0,1
0116,0000,0033,0507,0376,0164,0061,0352,0004,0189,0000,0000,a,0,0,0,1
0117,0009,0054,0508,0402,0165,0099,0397,0015,0190,0000,0000,a,0,0,0,1
0118,0002,0076,0507,0415,0163,0139,0429,0027,0190,0000,0000,a,0,0,0,1
0119,0013,0099,0507,0415,0162,0181,0420,0049,0189,0000,0000,a,0,0,0,1
0120,0022,0122,0508,0415,0162,0222,0413,0081,0189,0000,0000,a,0,0,0,1
0121,0052,0144,0507,0416,0161,0262,0391,0111,0190,0000,0000,a,0,0,0,1
0122,0079,0165,0507,0416,0160,0300,0369,0133,0190,0000,0000,a,0,0,0,1
0123,0116,0184,0507,0415,0159,0336,0338,0145,0190,0000,0000,a,0,0,0,1
0124,0145,0203,0507,0416,0159,0369,0316,0152,0190,0000,0000,a,0,0,0,1
0125,0171,0220,0507,0426,0159,0401,0316,0133,0189,0000,0000,a,0,0,0,1
0126,0196,0237,0507,0427,0159,0431,0298,0106,0190,0000,0000,a,0,0,0,1
0127,0216,0254,0508,0445,0161,0463,0318,0080,0189,0000,0000,a,0,0,0,1
0128,0232,0272,0507,0454,0161,0495,0323,0057,0189,0000,0000,a,0,0,0,1
0129,0231,0290,0507,0455,0161,0527,0326,0048,0189,0000,0000,a,0,0,0,1
0130,0229,0308,0507,0467,0161,0560,0351,0046,0189,0000,0000,a,0,0,0,1...
```

Figure 5-20:Format of the RS232 data collection and typical terminal log showing the data stream

- Diagnostic. This was a special mode for debugging interfacing problems. The flow signal to the DDP controller was brought to zero to disable the pump. Raw data from all the A/D converters and the digital inputs were transmitted over RS232 at 11Hz, while pulsing the digital outputs in sequence.

5.3.4 Development of control features

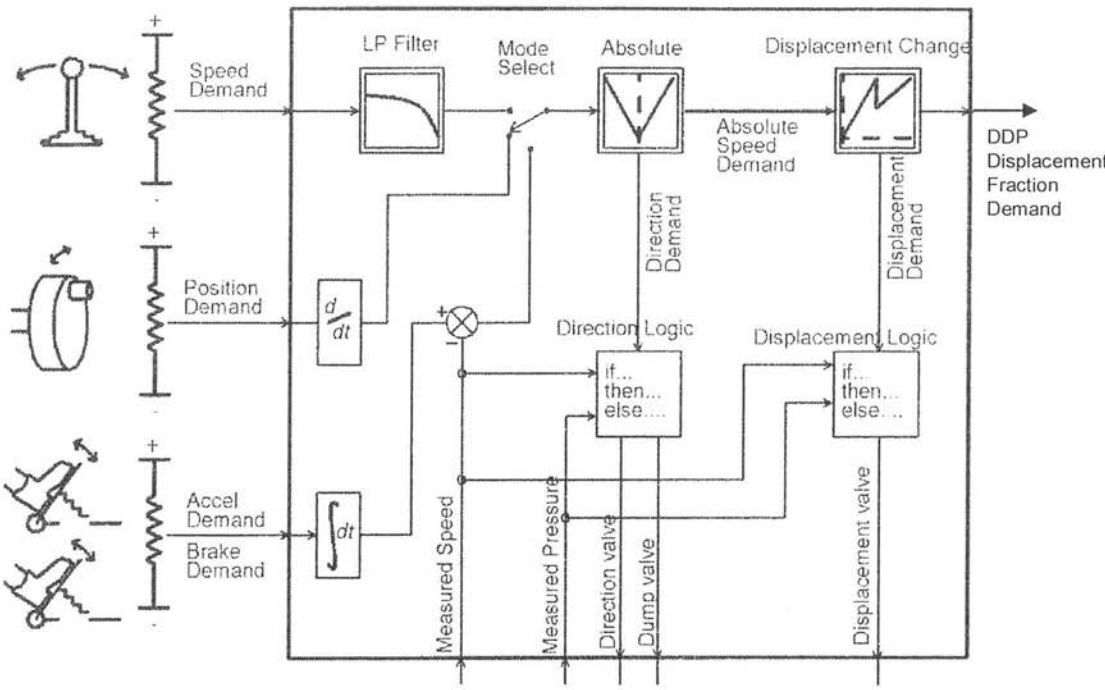


Figure 5-21: Simplified schematic of the system controller software

The system controller was based on an Intel 196 microcontroller which executed the control calculations at 100Hz. The simplified signal path is shown in the diagram above.

Every 10ms, the following sequence took place:

1. Sample all analogue and digital inputs
2. Decide on control mode – Idle, Position, Acceleration or Speed
3. Calculate pump flow demand from control mode
4. Decide whether to change direction valve; if so start changeover sequence
5. Decide whether to change displacement valve; if so start displacement change sequence
6. Decide whether stiction and leakage are preventing a smooth start; if so start the "jerk start" sequence
7. Send up to ten buffered characters to RS232 port
8. Wait for next time step

The control code was written in the C language; the source code is listed in Appendix 9.8.

The driver was able to control the vehicle with pedals, a joystick or a handwheel as shown below:

Control Mode Logic

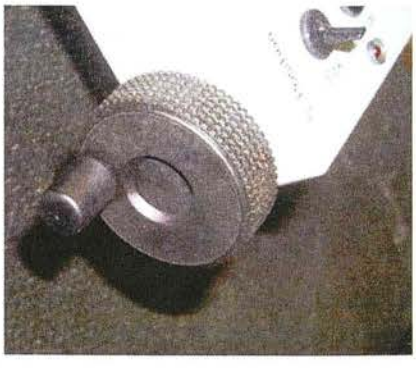
- At zero speed with no control input the system is prepared to accept an acceleration, speed or position demand
- Once a control is moved the controller will stay in that mode until the vehicle stops again
- The brake pedal always has priority!

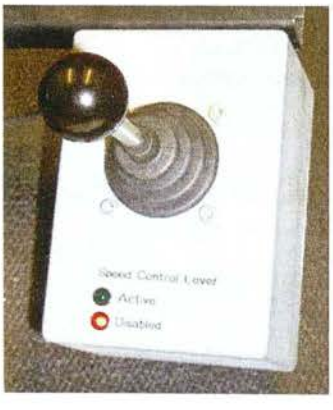
Accelerator and brake pedals

- Command acceleration proportional to pedal position, offset by some speed feedback for easier speed regulation
- The pedal signal is used as a flowrate rate-of-change of command
- Direction change by switch on control panel
- First 70% of brake travel commands hydraulic braking; the final portion includes friction brakes
- "Relaxed" driving style, similar to driving an car with automatic transmission

Position Handwheel

- Commands a bi-directional fluid volume
- Open loop "position control" with no position feedback
- Low and high range
- Volume resolution 0.34cc gives theoretical vehicle position resolution of 4mm
- Turns the motor into a hydraulic "stepper motor" with 28 steps per rev.





Speed Lever

- Commands a bi-directional flowrate, proportional to lever position.
- Selectable speed range for good low-speed control ("creep mode")
- Switchable low pass filter reduces control bandwidth for smoothness
- Similar to electrohydraulic swashplate control

Figure 5-22 The driver interface for the propel demonstrator vehicle

Acceleration control

Description

A linear potentiometer was fitted to both brake and accelerator pedals to provide voltage signals to the embedded controller. As the engine speed was governed to a fixed speed, the accelerator pedal only provided an electrical input to the system controller; the engine torque was adjusted automatically by the mechanical governor. Originally, the brake pedal actuated the rear drum brakes via a Bowden cable, and it was desired to retain this for safety. The cable was adjusted so that the mechanical brakes were only engaged when the brakes were depressed beyond 2/3 travel. In the first 2/3 of travel, the embedded controller produced hydraulic braking by ramping down the DDP displacement demand, causing the overcentre valve to provide back-pressure on the motor. Logic was added to the control code to ensure that if both brake and accelerator pedals were depressed, the brake signal had priority.

The aim was to emulate the easy driving feel of a vehicle with a torque converter, whereby the accelerator pedal commands torque at the wheels, rather than speed. Ideally this would be implemented by sending a pump pressure demand signal, proportional to the accelerator pedal signal, to the DDP controller. However the single analogue output of the system controller was already used for the displacement demand signal, so there was no way for the master controller to provide a separate pump pressure demand to the DDP controller. The second best option was to interpret pedal signals as a rate-of-change of speed command, with the accelerator producing an increase in flow-rate, and the brake pedal producing a decrease. The rate-of-change of speed demand was integrated by the vehicle controller to produce the speed demand, and hence the flow-rate produced by the pump.

In practise this pure integrator gave a disconcerting driving feel. With the accelerator pedal depressed to only 5%, the vehicle would accelerate at a constant rate until it eventually reached maximum speed. To better emulate the behaviour of a vehicle with a torque converter transmission, speed feedback was introduced to offset the zero point as a function of speed. This meant that as the speed increased, the pedal position required to maintain a constant speed, also increased. The effect of this was that with a step accelerator demand, the vehicle speed would asymptote towards a steady speed. This behaviour is shown graphically below:

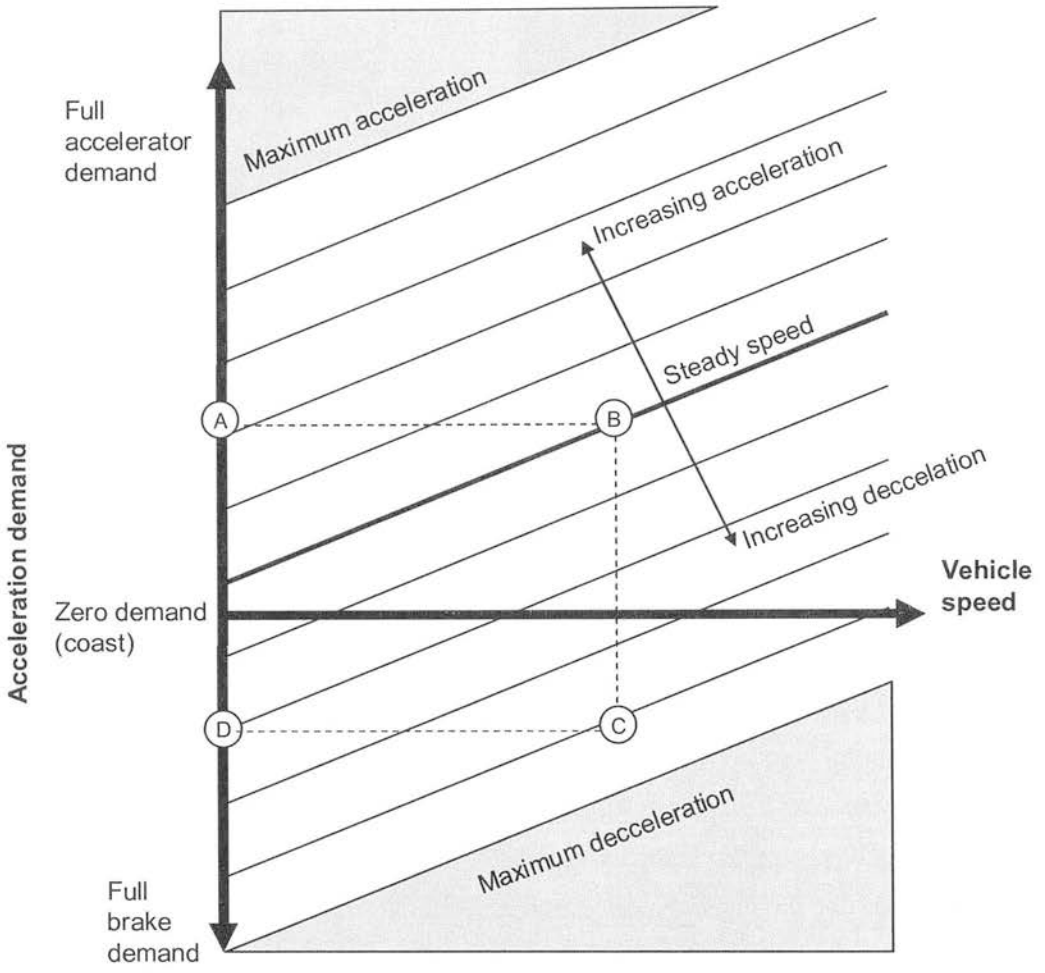


Figure 5-23: Acceleration mode: control function

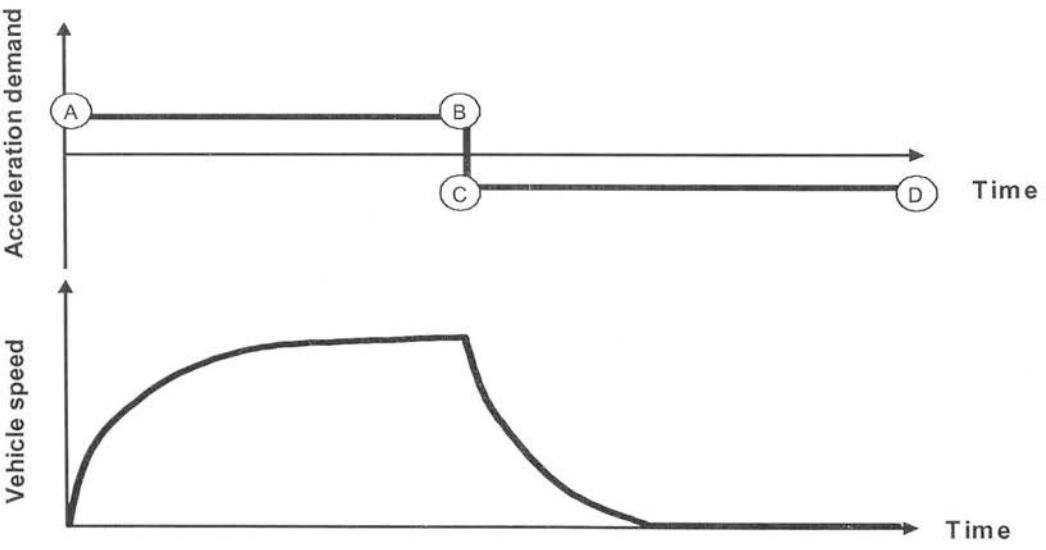


Figure 5-24: Acceleration mode: expected behaviour

Acceleration control algorithm

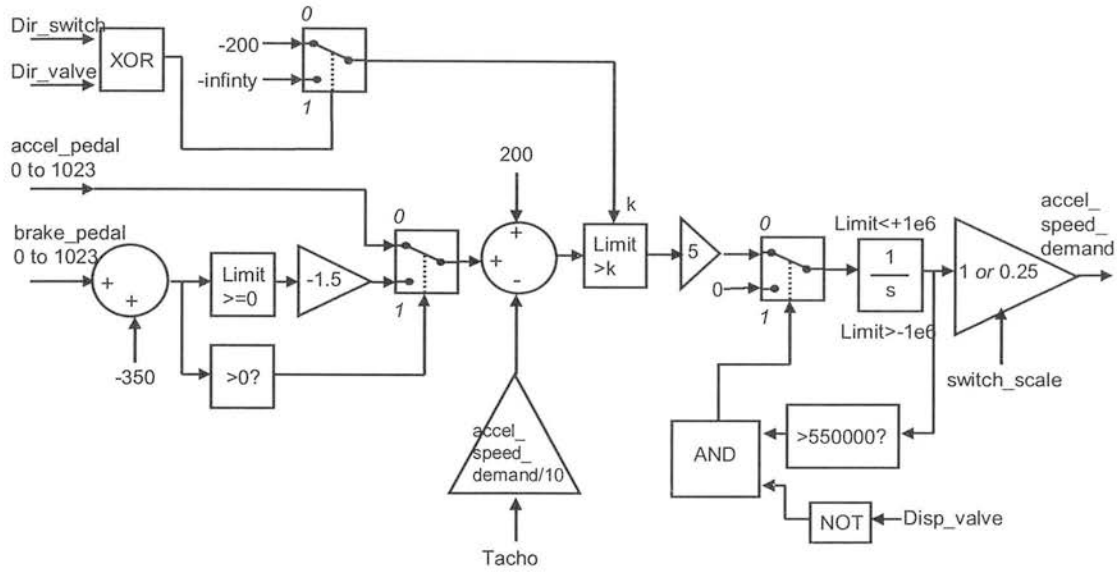


Figure 5-25: Acceleration control algorithm

The combination of accelerator and brake signals created a bipolar acceleration demand. This was offset by the vehicle speed signal from the tacho, and a constant to ensure that when both pedals were released the vehicle decelerated to zero speed in a finite time. If the direction selector switch was changed while the vehicle was in motion, deceleration occurred with a minimum rate even if the accelerator was still depressed. This acceleration signal was integrated to create a vehicle speed demand. Integration was temporarily suspended if the change to low displacement was in progress, to prevent integrator 'wind-up'. The speed demand signal was scaled according to the position of the 'scale' switch.

Note that in Figure 5-25 above, and Figure 5-26 and Figure 5-28 below, the functions of differentiation, integration and low-pass filter are represented by their equivalent Fourier transforms; these were in fact implemented as discrete-time functions, as the controller worked with a fixed sample rate.

Position control

The DDP is fundamentally a metering pump. Each unit of fluid volume delivered to the load is explicitly commanded by the microprocessor. At a simple level, fluid volume displaced into the motor corresponds to vehicle position, so the opportunity presented itself to demonstrate the metering capability of the pump in the form of an open-loop control of vehicle position. The ability precisely to control pump flow close to zero displacement allows the generation of smooth s-curve motion profiles.

Position control algorithm

A handwheel turned a potentiometer capable of continuous rotation. With a constant rate of rotation this created a saw-tooth voltage input to the controller. The voltage difference turned in 10 time steps was calculated to measure the speed of rotation, "un-wrapped" to take into account the saw-tooth pattern. This speed demand was filtered with a first order infinite impulse response (IIR) digital filter with variable cut-off frequency to produce a filtered speed demand signal. This was then fed into a summing junction, limited integrator and gain block as shown in the diagram below.

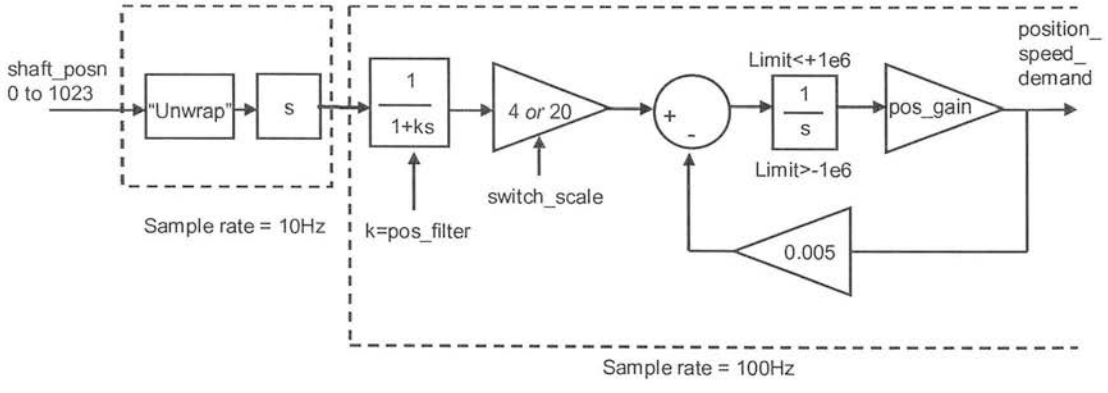


Figure 5-26: Position control algorithm

The result of this signal path is that a step speed input on the handwheel causes a flow demand profile to be produced which is continuously differentiable. By changing the feedback amount and the filter setting, smooth "s-curve" motion profiles were generated. The results of a simple simulation of this algorithm below show the smooth motion profile generated from a step input from the handwheel:

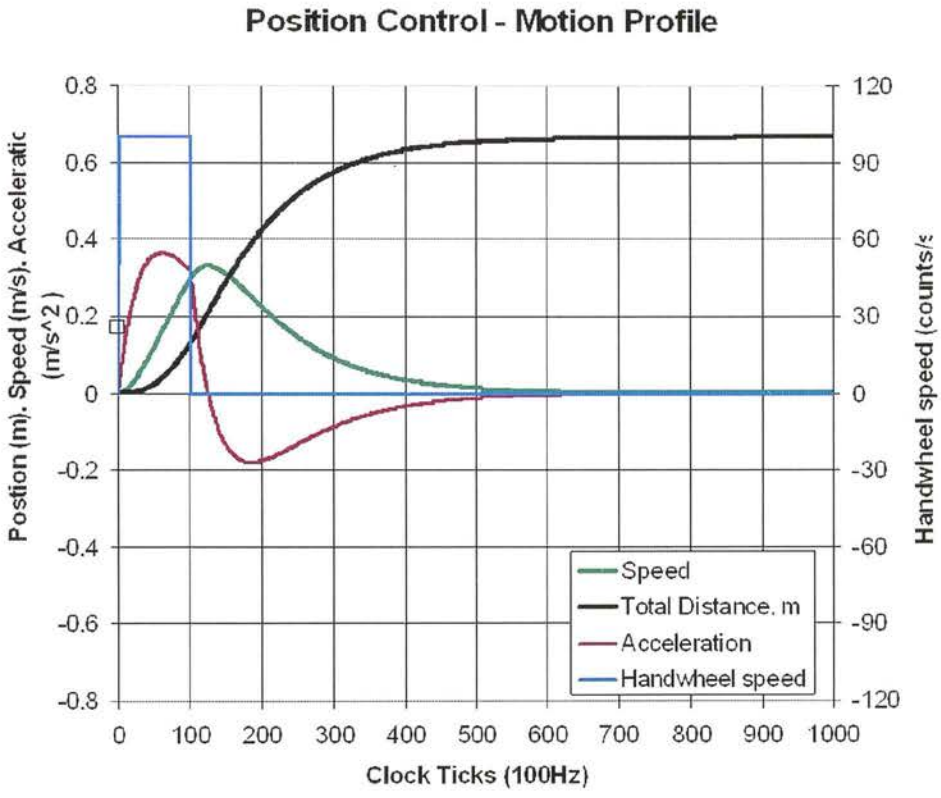


Figure 5-27: Simulation of the position control function, showing continuous acceleration with a step of handwheel rotation speed

Speed control

The speed control was open-loop; the vehicle speed sensor was not used. This is similar to a conventional electronic displacement control of a swashplate pump.

Speed control algorithm

A joystick was fitted with a centre detent. A deadzone allowed a clear neutral zone to be defined. This signal was then scaled and filtered with a first-order IIR low pass filter, the time constant of which was selected by the filter switch, to create a filtered speed demand signal. This filter could be selected by a switch between $\tau \approx 2$s ("loose" mode) and $\tau \approx 0.02$s ("tight" mode).

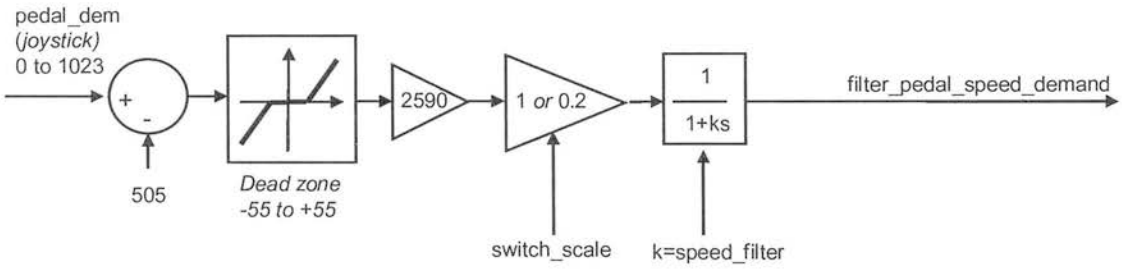


Figure 5-28: Speed control algorithm

Motor displacement switching

All three of the control modes produce a speed demand of the vehicle. The flow command to the pump was product of the speed command and the present motor displacement, which was switchable between 9.5cc/rev and 5.5 cc/rev. The vehicle always started with the motor set at full displacement. As the vehicle speed command increased, the flow command to the pump increased in proportion.

For the controller to decide to change to reduced motor displacement, a number of conditions had to be satisfied:

- the speed demand must exceed 54%;
- the actual speed must exceed 45% of maximum;
- the pressure must be less than 120 bar.

A pressure threshold acted to inhibit a change to reduced displacement when going up hill, when it was sensed that there would be insufficient wheel torque to continue with reduced displacement. The prevented "hunting" behaviour.

Once the decision was taken, the displacement change valve was energised and a timer started. Once this timer expired, the displacement change was assumed to have taken place and the DDP displacement was reduced in a sudden step to the multiple of the speed demand and the reduced motor displacement.

The change back to full displacement was made when the following conditions were satisfied:

- the speed demand must be less than 48%;
- the actual speed must be less than 40% of maximum.

There is a difference between the thresholds to initiate reduced displacement, and to initiate full displacement. This difference was arrived at by tuning parameters while driving, to prevent rapid switching between displacements when the speed demand was in the change-over region.

The motor displacement change was a discontinuity from the hydraulic point of view, causing step changes in flow and pressure. However it was found that the displacement change of the pump and motor could be synchronised so that the change was transparent to the driver. Results which demonstrate this are shown later.

Stiction compensation

In flow-controlled hydrostatic systems, leakage and stiction can combine to produce jerky starting of a vehicle. Normally the displacement demand of the pump increases linearly with speed demand, so the effect will be that the vehicle will not move until the displacement fraction of the pump is high enough to produce the breakaway pressure across the internal leakages of pump and motor, that pressure being sufficient both to overcome internal motor friction and whatever external load is applied – typically pushing the vehicle up a hill. Once this critical pressure is produced, the vehicle starts to accelerate. However as soon as the motor moves, the stiction reduces, so more accelerating torque is produced by the motor. At this point the flow taken from the pump by the rotation of the motor is insignificant compared to the leakage flow which was required to break away. Even if the displacement fraction of the pump is held constant, the vehicle accelerates until the flow taken by the motor equals that supplied by the pump minus the leakage flow in both pump and motor at this new lower pressure. With a slow speed demand ramp, the vehicle will not move for a significant time, and then suddenly jump to a non-zero speed.

At start-up the displacement fraction of the pump is very low. Conventional variable-stroke hydraulic machines have leakage which is independent of the displacement fraction (Dorey 1988). This means that internal leakage in the pump is similar at start-up to what it is at high vehicle speed. Using a DDP as a propel pump may offer a significant advantage because in

the DDP, internal leakage reduces as displacement fraction reduces. This means that at start-up the internal pump leakage becomes insignificant compared to the motor leakage, so the breakaway pressure can be generated at a lower flow-rate compared to a conventional variable-stroke pump. This means that the jerk at start-up should be significantly reduced.

Even with this effect in its favour the vehicle still exhibited some jerk when starting on a hill, when the fluid was warm (>50°C). A technique was developed which exploits the response speed of the pump to create a sudden step demand of 100% for 100ms, if stiction is detected; this is sufficient to get the vehicle moving whereupon the displacement returns back the that demanded by the operator. Results from this "jerk start" algorithm are shown in Section 5.4.2 below.

5.3.5 Data acquisition

The data transmitted by the controller was limited by capacity of the RS232 interface to an 11Hz update rate, which is too low to capture the detailed dynamics of a hydraulic system. To provide this high bandwidth information, a separate 16-bit DAQ was used[3] to acquire the following signals:

- Pump pressure transducer (Keller 0-250 bar, 0-5V).
- Tachogenerator driven by motor pulley (Maxon DC motor acting as a generator).
- Displacement demand voltage transmitted by system controller to the DDP controller.
- Trigger pulse signal (one pulse per rev) from DDP crankshaft sensor – giving engine speed.
- Current measurement from a Hall-effect DC current clamp around the DDP supply cable - capturing a pulse every time a LPV is closed.
- Accelerometer; +/- 1g silicon strain-gauge type with internal signal conditioning, DC to 5kHz response; attached to the vehicle chassis.
- Position signal from a potentiometer attached to the vehicle, turned by a steel wire wrapped around a spring-loaded drum of constant radius (commonly called a "yo-yo" sensor), one end of which was fixed to a stationary object. The full range of the potentiometer corresponded to 6m of vehicle movement.

---

[3] Data Translation DT9804, 16 single-ended analogue inputs +/-10volt, 16-bit, USB2.0 connection to PC

In retrospect, some errors were made to the instrumentation which hampered later attempts to analyse the results:

- A pressure transducer should have been fitted on the up-stream port of the overcentre valve. This would have indicated the back-pressure applied to the motor during braking, giving an important insight into the dynamic operation of this valve.
- The accelerometer was rigidly mounted to the chassis of the vehicle, causing the high-frequency vibrations of the engine to overwhelm the much smaller low-frequency vehicle accelerations. Mounting the accelerometer with a rubber isolator would have provided more useful data.
- Sensing engine speed once per revolution provided insufficient resolution and sample rate; a tachogenerator or optical encoder would have provided more useful data.

Before tests began, a sample rate had to be chosen which guaranteed that the important information was captured, without generating an inconvenient amount of data by using too high a rate, or causing signal distortion due to aliasing by sampling too slowly. Each channel was subject to a test whereby the sampling rate was decreased in steps, from the 25kHz maximum of the DAQ, until the rate was reached at which aliasing could be observed in the sampled data stream when compared to an oscilloscope trace. As a result of these tests 2kHz was chosen for all of the tests. To reduce the volume of data for subsequent analysis, this was re-sampled to 1kHz using a moving-average filter.

5.3.6 Data processing

The DAQ data and the RS232 data have different characteristics.

The RS232 data from the controller records both physical measurements from the system controller ADC and internal software variables, but is limited to 11Hz sample rate. The RS232 data is mainly useful for an insight into how the system control algorithms are working.

The DAQ data records high-bandwidth signals to do with the engine, DDP and the hydraulics system, and is mainly useful for an insight into detailed system dynamics.

The following processes were applied to the DAQ data in a post-processing spreadsheet:

Pressure signal – scaled into bar with the manufacturer's supplied linear calibration.

Tacho – scaled into vehicle speed in m/s. The scale factor was calibrated by experiment, comparing the voltage with motor speed, measured with an optical tachometer. Knowing the drive ratios of the belt and differential, and the effective rolling radius of the wheel, this allowed the tacho voltage to be scaled to vehicle speed.

DDP displacement demand signal – this 0 to 5V signal corresponded to the square root of displacement demanded; this was converted into a percentage displacement demand.

Trigger pulse – this is the short pulse from the shaft position sensor on the DDP, and was used as a measure of pump/engine shaft speed. Ideally the period of this signal would be measured with a clock-based digital electronic circuit, but this feature was not available on the DAQ used. The pulses were therefore sampled with the analogue data, and a post-processing routine was written to measure the time between trigger edges in the sampled data and therefore calculate engine rpm. At 1500rpm, the average period of the pulses was 40ms, or 80 samples of the DAQ; therefore this measurement was highly quantised. For some subsequent analysis, this raw period measurement was post-processed with a moving average filter with a rectangular window of +/-200ms to create a smoother estimate of engine speed, as shown below.

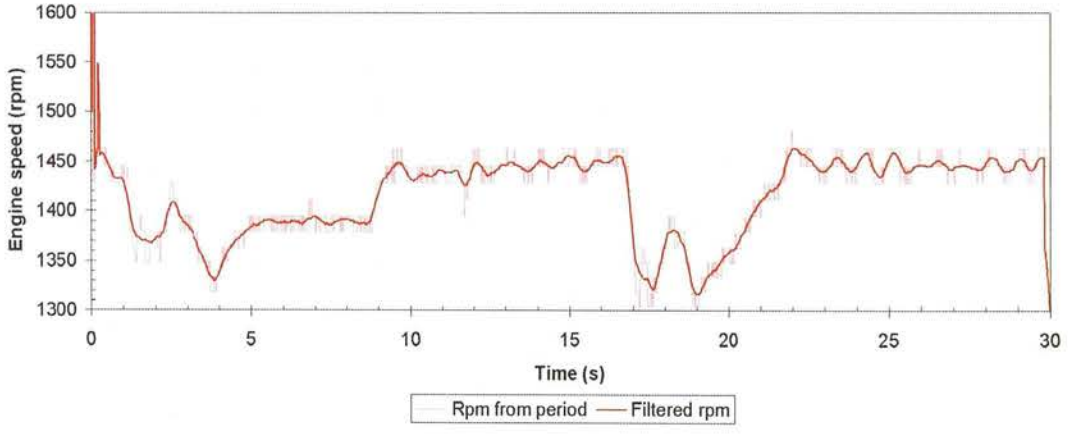


Figure 5-29: Test 1-3; measurement of engine speed by trigger pulse period

Current measurement – this was averaged to create a measurement of the actual enabling fraction of the pump, on the assumption that each LPV pulse from the DDP resulted in the same electrical charge being transferred from the battery to the coils; this is valid only as long as the pulse width is constant (controlled by a 555 monostable timer), the coil temperature is constant, and the motion of the LPV is consistent.

Accelerometer – this was linearly scaled to ms^{-2} according to the manufacturer's datasheet.

5.4 Experimental results

A range of tests were carried out to evaluate the behaviour of the vehicle. The maximum test length was restricted to 30 seconds to keep the data files to a manageable size – 30,000 samples of each channel. Each test was designed to investigate the behaviour in a particular mode of operation. All tests took place on flat ground (unless otherwise noted) after a 10 minute warm-up drive.

Number	Mode	Speed range	Control	Description
1-1	Accel	Fast	Loose	Gentle acceleration to full speed, gentle braking to stop
1-3	Accel	Fast	Loose	Medium acceleration to full speed, medium acceleration to stop; repeated at full acceleration
2-3	Speed	Fast	Loose	Four speed steps to maximum
2-4	Speed	Fast	Tight	Four speed steps to maximum
2-5	Speed	Fast	Tight	Speed steps; motor at full displacement only
2-6	Speed	Fast	Tight	Speed step to maximum, forward and reverse
6-2	Speed	Slow	Loose	Starting on a hill, with and without "jerk start" routine active
7-1	Position	Slow	Loose	Forwards and backwards position steps
7-4	Position	Fast	Loose	Forwards and backwards position steps
7-5	Position	Fast	Tight	Forwards and backwards position steps
8	Accel	Fast	Loose	Effect of displacement change delay
9-5	Position	Slow	Loose	Forwards and backwards position steps – DDP part strokes disabled
9-6	Position	Slow	Loose	Forwards and backwards position steps – DDP part strokes enabled

5.4.1 Synchronising serial and analogue data

Firstly, the results from test 1-1 are examined, a gentle sequence of accelerations and decelerations throughout the whole forward speed range, on flat ground. The RS232 data from this run is shown below:

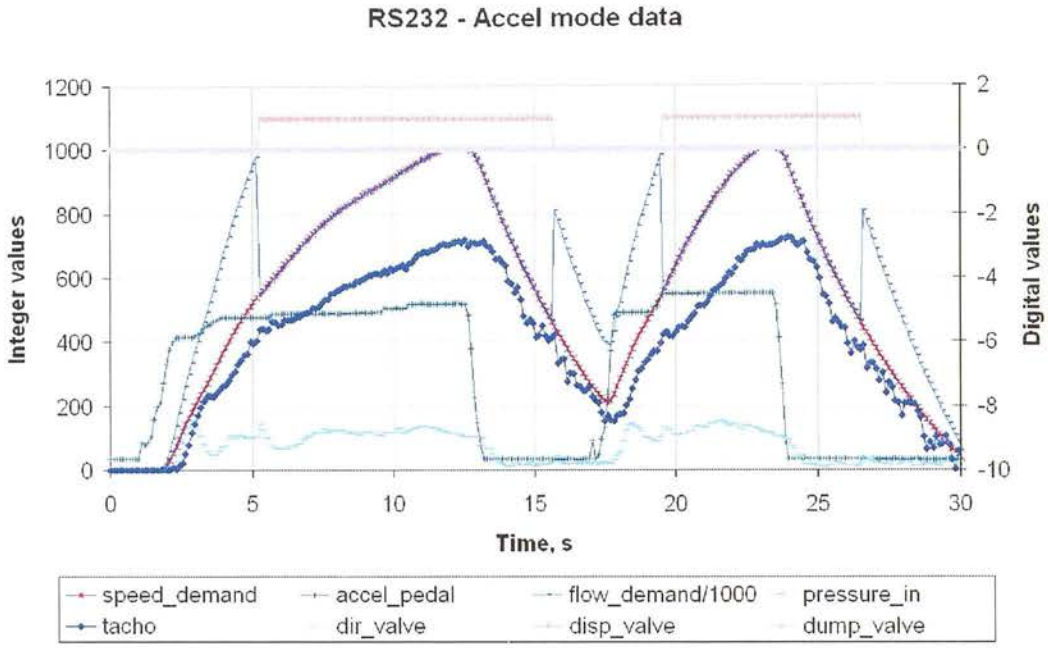


Figure 5-30: Test 1-1; RS232 data

The trace of *accel_pedal* shows that the driver gave a fairly constant demand during the initial acceleration, and that this resulted in the expected asymptotic behaviour of the speed demand. The control logic of the displacement change can be clearly seen, with the sudden step in displacement demand to the DDP at t=5s at the same time as the direction valve is energised. The system controller estimate for the motor displacement when switched between displacements is obviously fairly accurate because the tacho signal is fairly continuous through this sudden change. The tacho signal follows the form of the *flow_demand* signal through deceleration, indicating that the overcentre valve is working to brake the vehicle when the pump flow reduces.

The RS232 and the DAQ data were synchronised by applying a time offset to the RS232 data. In the chart below, the lines show the DAQ data at 2kHz, while the points show the RS232 data at 11Hz.

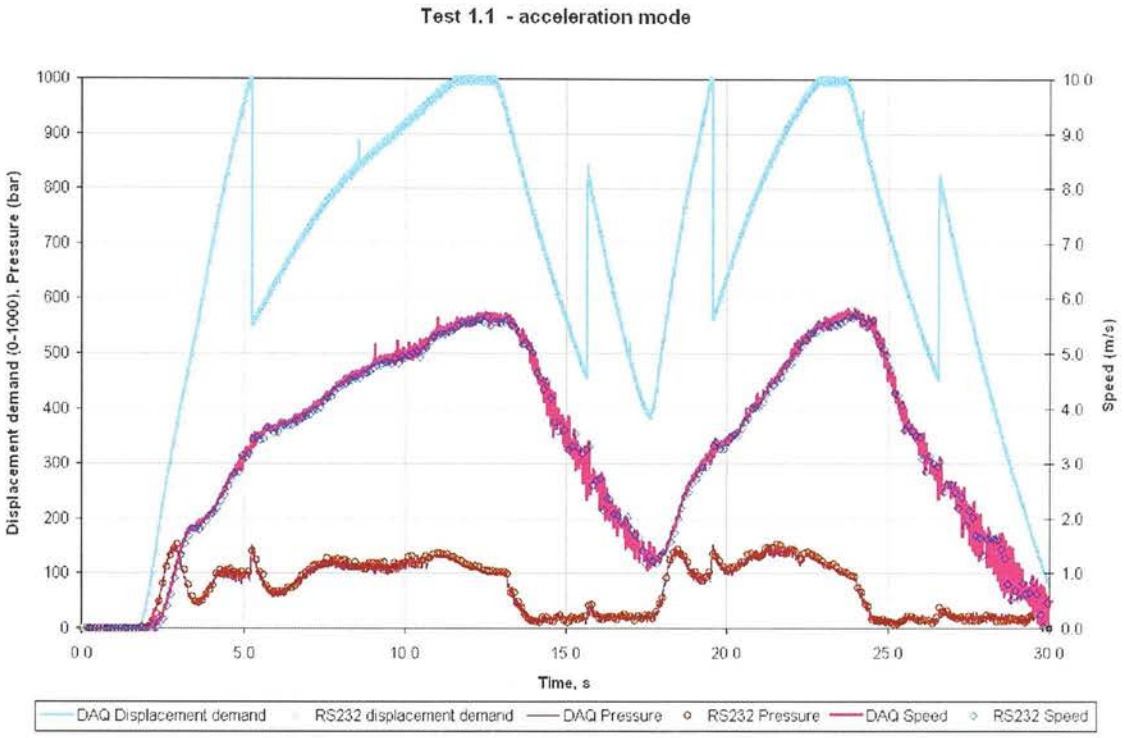


Figure 5-31: Synchronising the DAQ and RS232 data

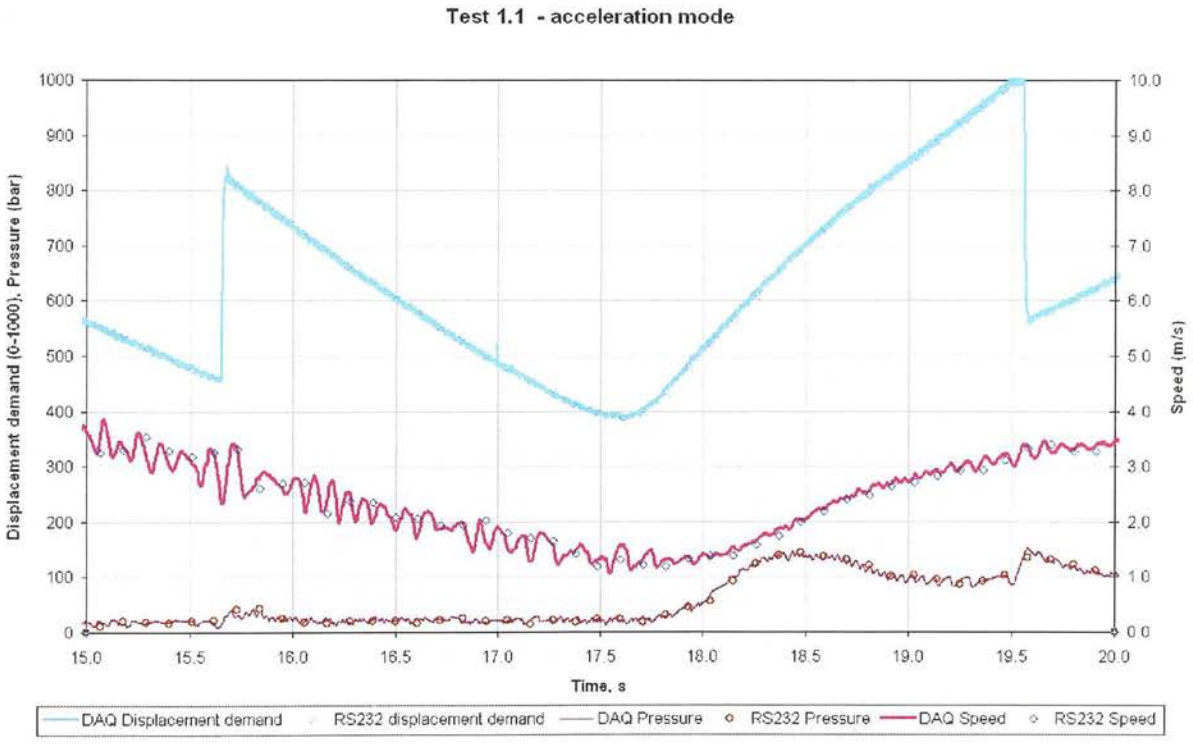


Figure 5-32: Detail of above figure

It can be seen that the data from the two sources can be synchronised, but the DAQ data is of much higher accuracy and sample rate – this is particularly evident when examining high-

frequency pulsation of the tacho and pressure signals. The DAQ data is used for all subsequent analysis.

Test results: Acceleration mode, test 1-3

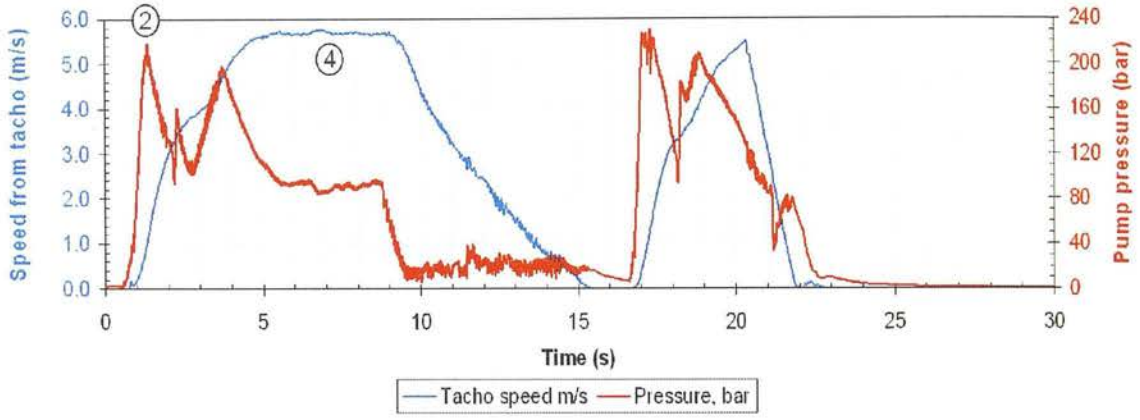


Figure 5-33: Test 1-3; pump pressure (red), motor speed (blue)

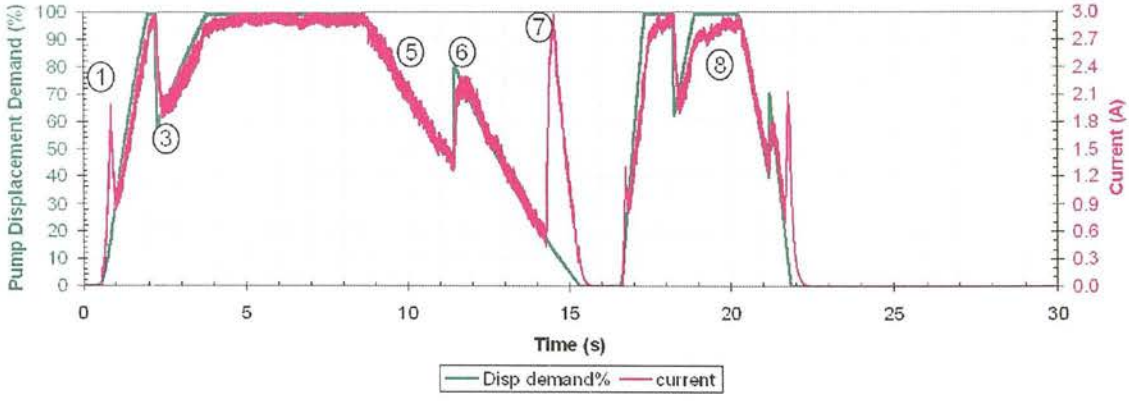


Figure 5-34: Test 1-3; Displacement demand voltage (green) and smoothed DDP current consumption(pink)

As each LPV pulse consumes the same amount of charge, the DDP current consumption is a good measure the frequency of LPV actuations. In the above trace, current was smoothed with a moving average filter (width 0.05s). It can be seen that the current, and therefore the frequency of LPV actuations, was linear with the displacement demand, subject to a step change in scale at the transition between part and full stroke modes.

Points of interest are numbered in the figure above:

1: The DDP changes from part stroke to full stroke mode, causing a step in the current trace.

2: The initial pressure peak is a function of the fundamental mechanical/hydraulic system dynamics.

3: The motor displacement changes to minimum, and the pump displacement reduces to compensate.

4: Steady state is reached at maximum flow and vehicle speed.

5: The pump displacement reduces; the overcentre valve creates a back-pressure on the motor causing deceleration. The pump pressure in this phase stabilises under 25 bar, this is the cracking pressure of the overcentre valve (200 bar) divided by its area ratio (8:1).

6: The motor is switched back to full displacement and the DDP simultaneously increases output flow to compensate; the speed trace is continuous, showing that the feedforward compensation of motor displacement is accurate.

7: The DDP transitions back to part stroke mode as shown by the step increase in current.

8: In the second phase, the acceleration is higher, placing a greater load on the engine. The current trace can be seen to droop from the displacement demand; this is due to the engine speed droop under load.

Engine governor behaviour

An estimate of the torque exerted by the DDP on the engine shaft can be made by multiplying displacement demand fraction, pressure and DDP maximum displacement. This is plotted below with the "droop" of the engine speed, being the difference between the no-load speed (1455rpm) and the measured speed:

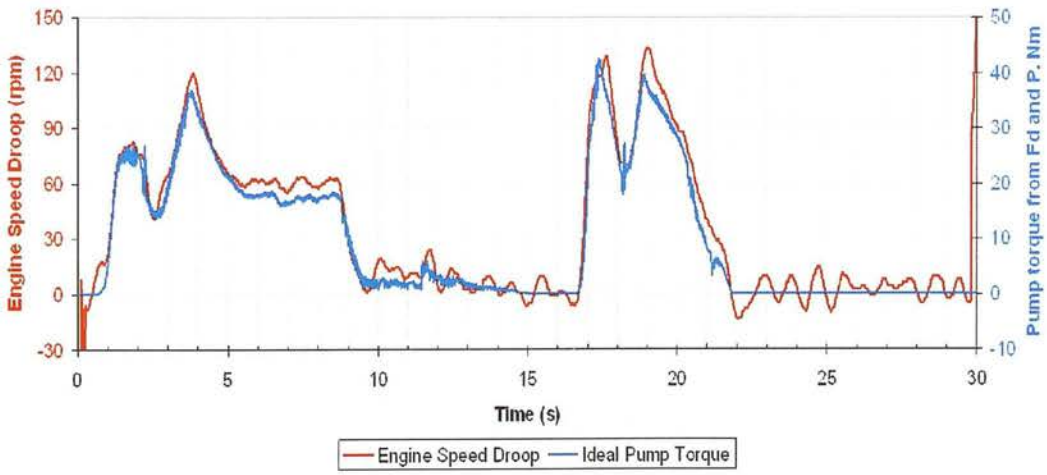


Figure 5-35: Test 1-3; Estimated pump torque (blue) and resulting filtered engine speed droop (red)

This data is shown below with torque on the x-axis.

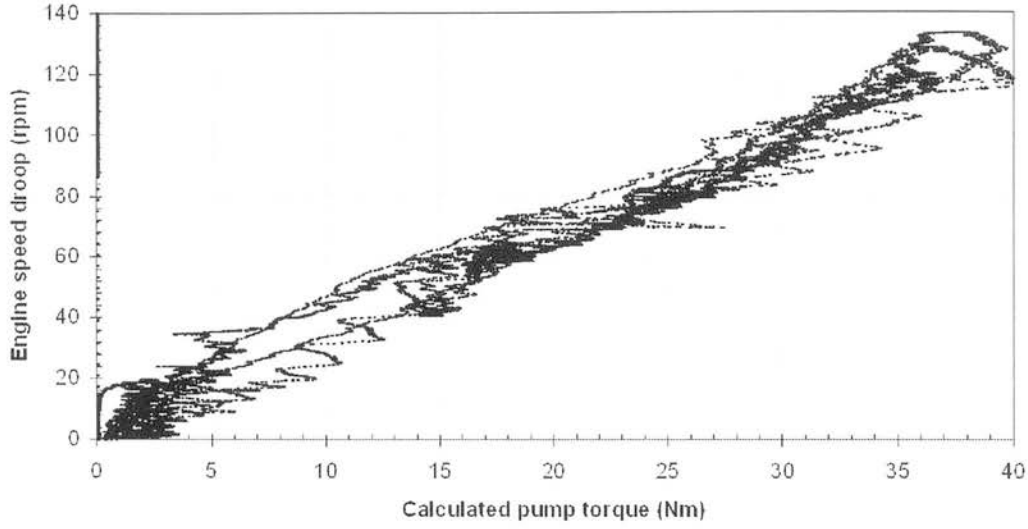


Figure 5-36: Test 1-3; correlation of calculated pump torque and engine droop

There is reasonable linearity between speed droop and torque, typical of an asynchronous governor. From the above graph it seems that the engine speed droops by 80rpm with 25Nm of torque applied – this value was used for later simulation work.

Details of motor displacement change event

The technique of switching motor displacement can only work without jerk if the feed-forward compensation delay applied to the DDP displacement command is accurately calibrated.

The functioning of the displacement change was investigated in test 8:

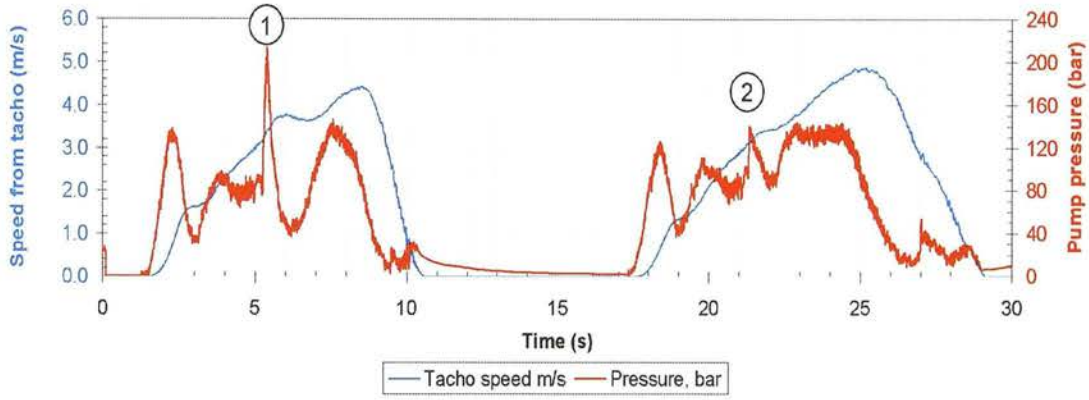


Figure 5-37: Test 8; effect of varying displacement change delay

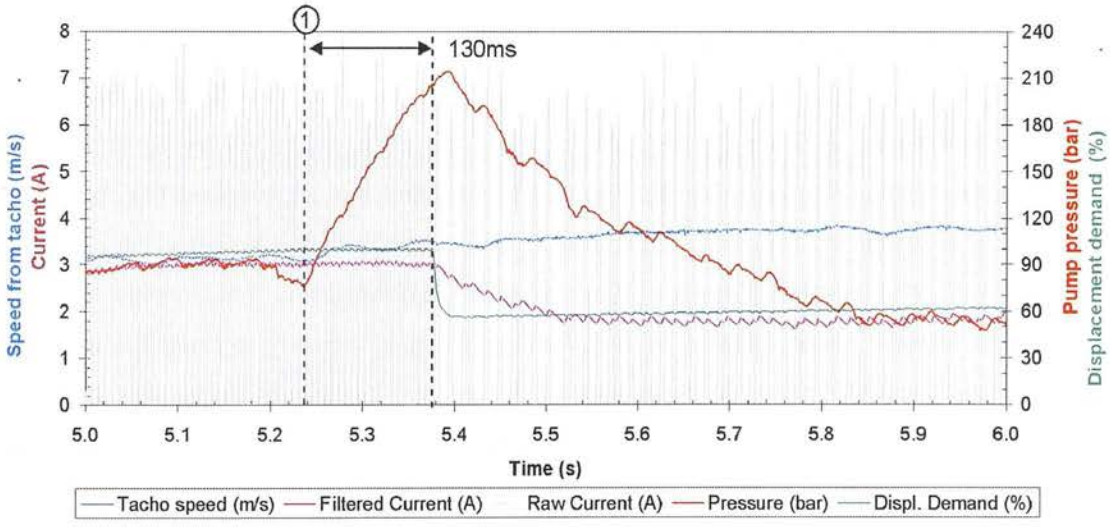


Figure 5-38: Test 8; detail of change to low displacement; delay too long

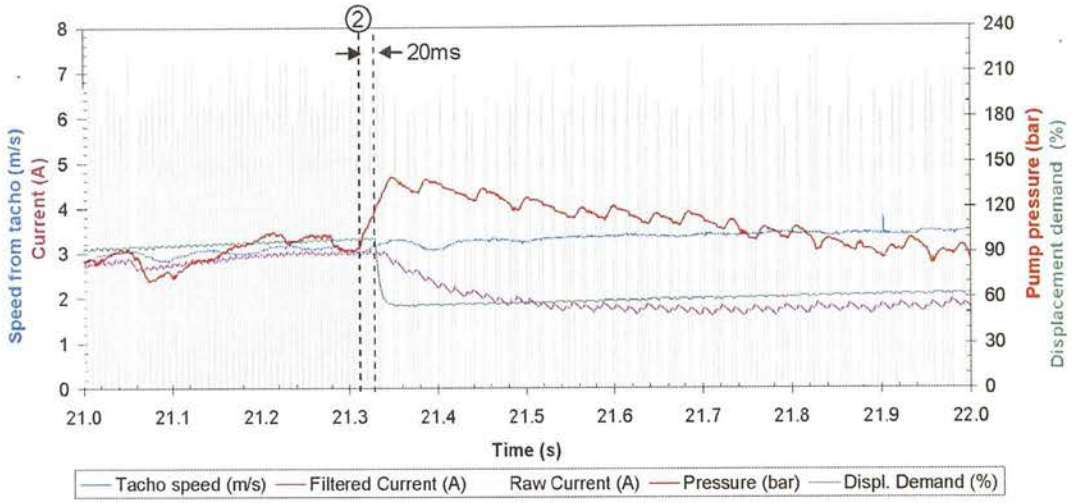


Figure 5-39; Test 8; details of change to low displacement; delay correct

The figures above show the displacement demand signal to the DDP and the resulting (unfiltered) current traces. Each current pulse corresponds to a LPV being actuated. It is apparent, from the trace of current, that the DDP reacts almost instantaneously (within 10ms) to the reduction in demand, by reducing the enabling fraction.

Test 8 consisted of two gentle accelerations and decelerations. For the first, the value of system controller parameter *low_to_high_delay* was set to 210ms. It is apparent (1) that the pressure starts to rise 130ms before the displacement of the DDP is reduced, indicating that the delay estimate is 130ms too long. An accelerating jerk could be felt by the driver due to this pressure spike. In the second phase of test 8, the delay was set to 100ms (2); this had the effect of reducing the width of the pressure pulse to 20ms, which eliminated the jerk. If the delay was set to less than 50ms (i.e. 30ms too short), a deceleration jerk could be felt during the displacement change, caused by the pressure falling sufficiently to cause the overcentre valve to create a momentary back-pressure on the motor. The value of 100ms was used for all subsequent tests because the deceleration jerk was more noticeable than the opposite case; variation of the actual actuation delay caused by fluid viscosity changes as the temperature changed was of the order of 20ms.

A similar experiment was carried out to establish the optimum delay in the case of switching from low to high motor displacement (controlled by parameter *high_to_low_delay*). In this case the optimum was 70ms; this is thought to be shorter because of asymmetry in the swashplate actuation mechanism. In this case an excessively long delay caused a decelerating jerk, while an excessively short delay caused an accelerating jerk.

Once calibrated, the displacement changes were not detectable by the driver, despite the sudden change of both flow and pressure.

5.4.2 Speed control mode

The speed control mode is the most direct form of control implemented and therefore reveals most about the underlying system dynamics. Two modes could be selected by means of the 'filter' switch. In 'tight' mode (Test 2-4), the input from the joystick was filtered with a short time constant ($\tau \approx 0.02$s) and therefore the driver directly controlled the speed demand. In 'loose' mode (Test 2-3) the input from the joystick was filtered with a much longer time constant ($\tau \approx 2$ s). For both tests the driver increased the speed demand in four steps from zero to full speed, and the motor displacement was fixed at maximum.

'Tight' Mode

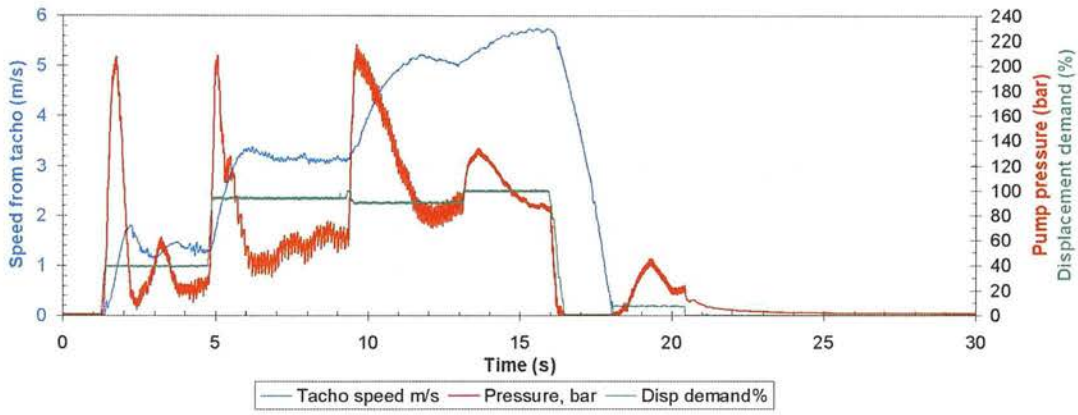


Figure 5-40: Test 2-4; Response of system to steps of DDP displacement

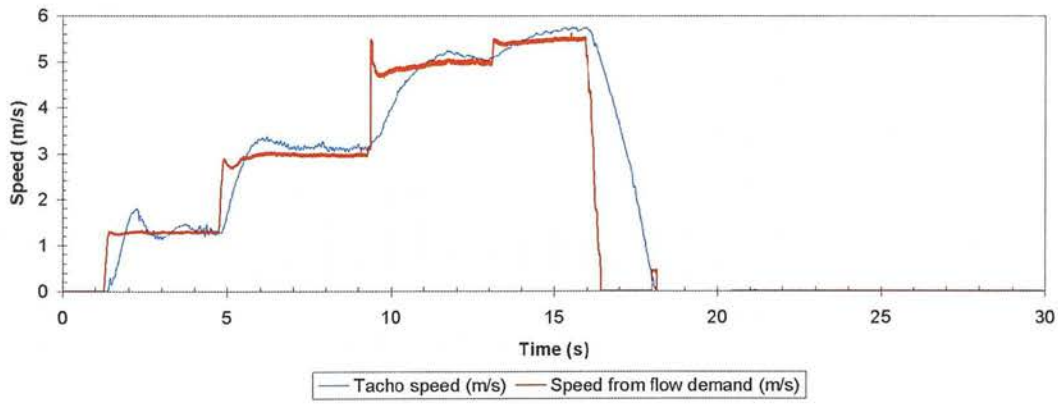


Figure 5-41: Test 2-4; actual speed and ideal speed (from displacement demand)

The figures above show that the underlying physical dynamics of hydraulic circuit dominate the behaviour of the vehicle when excited by steps of displacement demand. The smallest step causes a substantial overshoot in speed and the system is under-damped; as the speed increases, increasing leakage and pressure drop in the hydraulic system causes the speed response to become more damped.

A closer examination of the initial over-shoot from t=1s to t=5s reveals the natural frequency of the hydraulic/mechanical system:

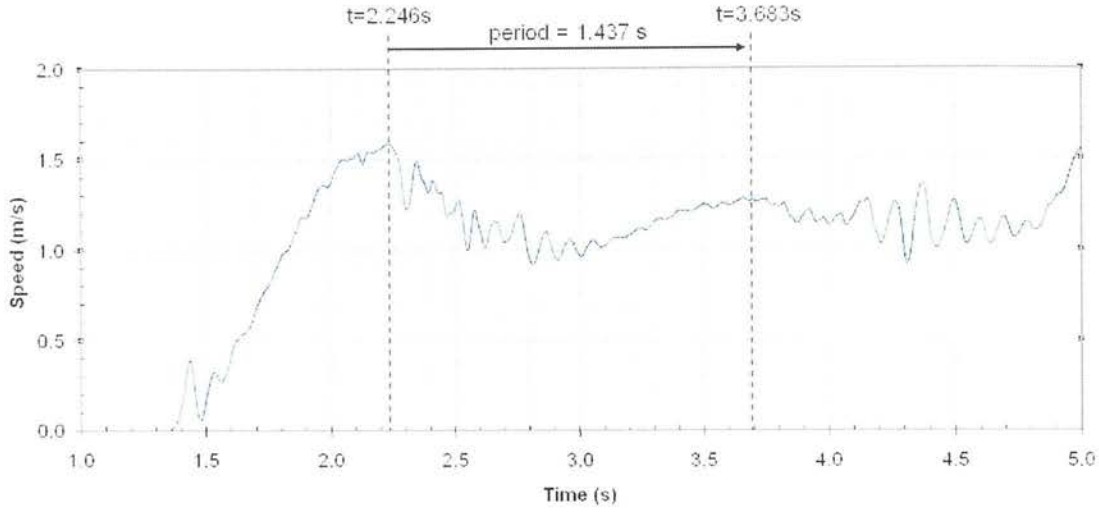


Figure 5-42: Test 2-4; motor speed period of oscillation

The period of oscillation of the initial over-shoot of motor speed to a step flow input was 1.44s. The undamped natural frequency expected from the simple linear analysis presented in Figure 6-8 (p.303) is 1.40s. This confirms that simple linear theory, neglecting the damping from leakage and stiction, can usefully be applied to predict the natural frequency of the mechanical/hydraulic system at low speed, when damping is low. However across the full speed range the system cannot be modelled as a simple second-order system with constant damping factor; this would exhibit the same over-shoot at all speeds, instead of the increasing damping factor with speed which was observed in experiment.

'Loose' mode

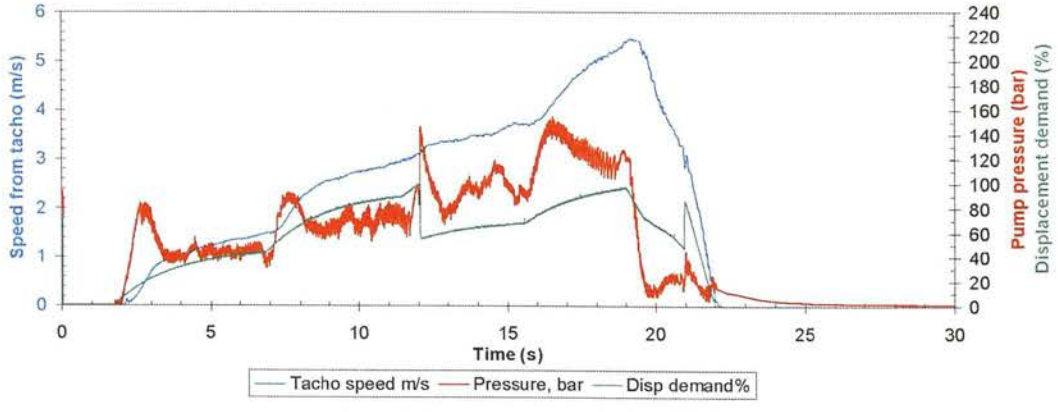


Figure 5-43: Test 2-3; effect of first-order filter; red=pressure, blue=actual speed, green=displacement demand.

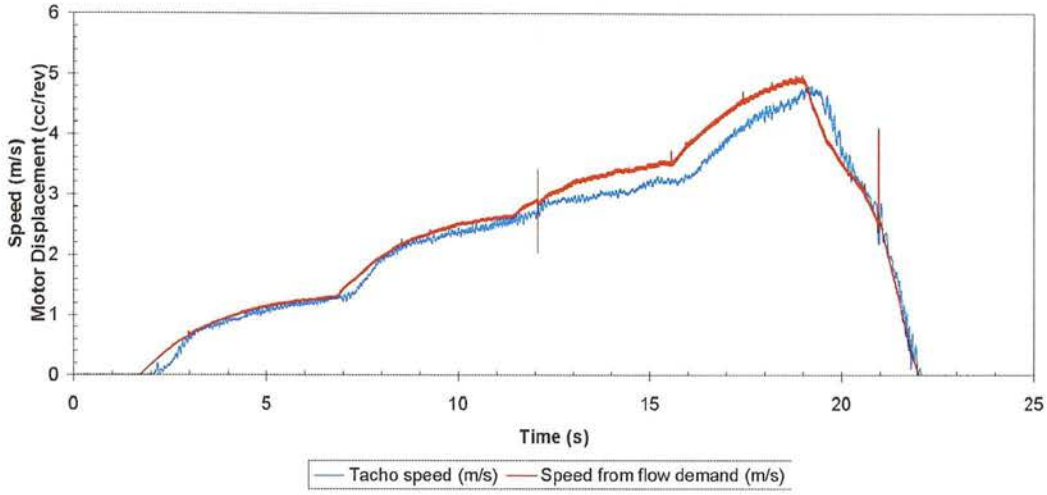


Figure 5-44: Test 2-3; effect of first-order filter; blue=actual speed, red=speed demand

The long time-constant filter on the input demand slows the response of the system to input steps, but has the desirable effect of hiding the underlying system dynamics from the driver. The actual speed follows the desired speed more closely than with the short time-constant filter.

Symmetry of behaviour under overcentre braking

Test 2-5 was a repeat of 2-4 with a decelerating step; this was included to verify that the system dynamics were symmetrical, whether accelerating or decelerating.

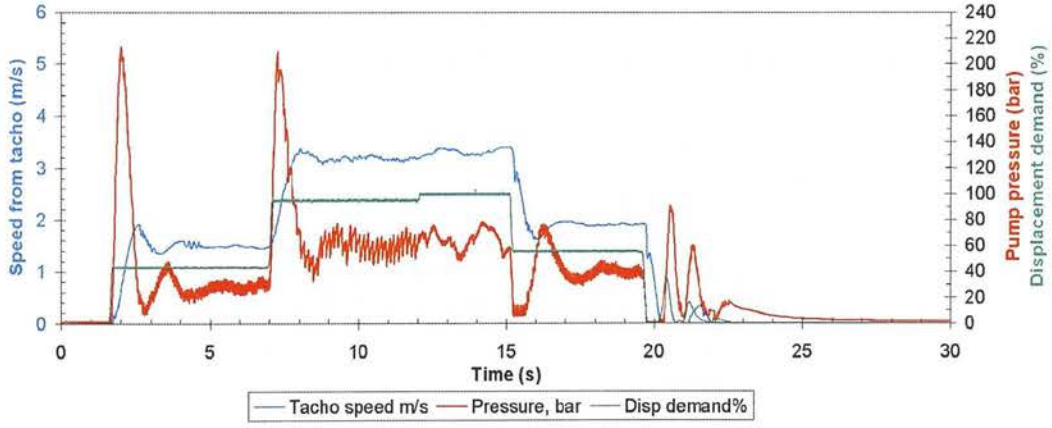


Figure 5-45: Test 2-5; symmetry of system response to speed demand steps

The speed under-shoot at t=16 is similar to the over-shoot at t=2.5. This confirms that the underlying system dynamics are symmetrical whether the speed demand step is positive or negative.

Pressure limiting and direction change

For most of the tests, the system pressure did not exceed the pre-set DDP software pressure limit of 200 bar, so the flow produced by the DDP closely followed the demand. However, with the speed demand filter in 'tight' mode, a sudden demand of full speed cannot be satisfied by the system without hitting this limit. This allows the operation of the software pressure limit to be examined. This test also featured a sudden reversal of speed demand from forward to reverse, allowing the direction change to be examined.

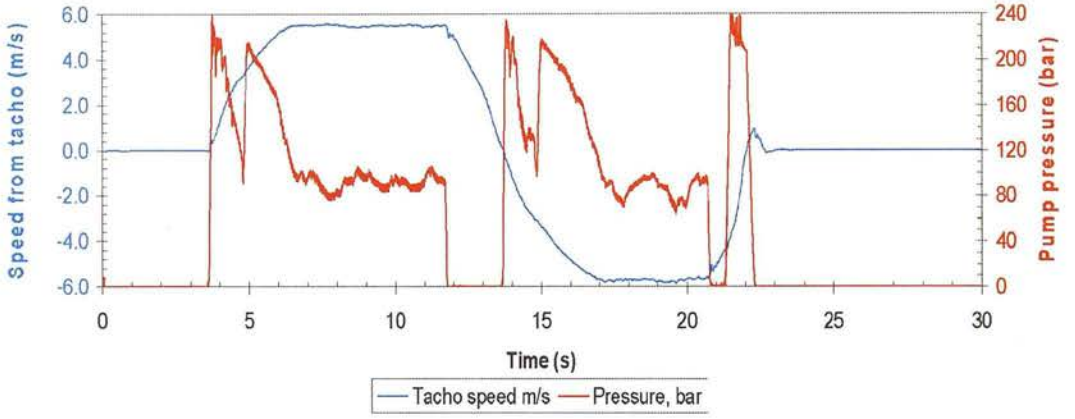


Figure 5-46: Test 2-6; step demand to full positive and negative speed demand; red=pressure, blue=motor speed

It can be seen from the trace of motor speed that acceleration was continuous through zero speed (t=13.7s). The coordination of the direction changeover valve and the DDP displacement seems to be accurate, allowing continuous control though zero speed much like a closed-circuit transmission with an over-centre swashplate pump.

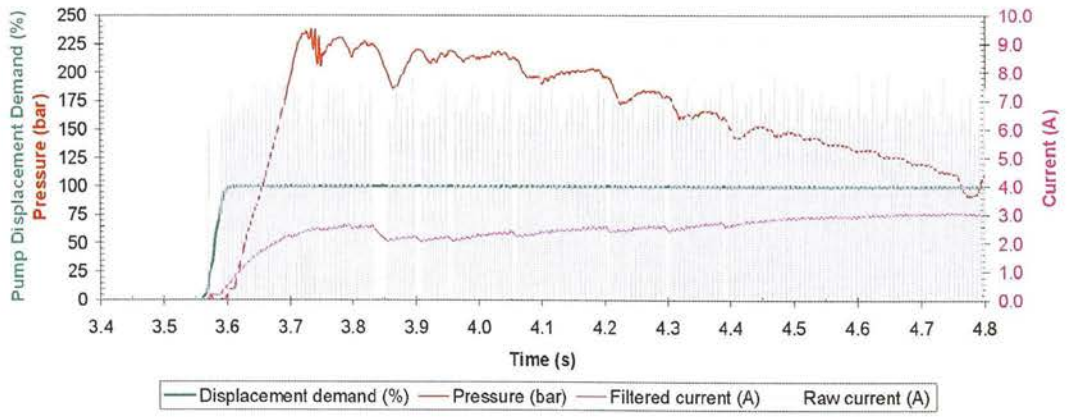


Figure 5-47: Test 2-6; software pressure limit in action

The above figure shows that the displacement demand during the initial acceleration saturated quickly to 100% at t=3.6s, and the DDP pumped at full displacement. The pressure rose to 227 bar at t=3.7s; at this point the gaps in the current trace show that the DDP started to introduce idle strokes as the pressure limiting algorithm came into effect. By t=4.4s the DDP was back to full displacement.

Stiction compensation ('jerk start')

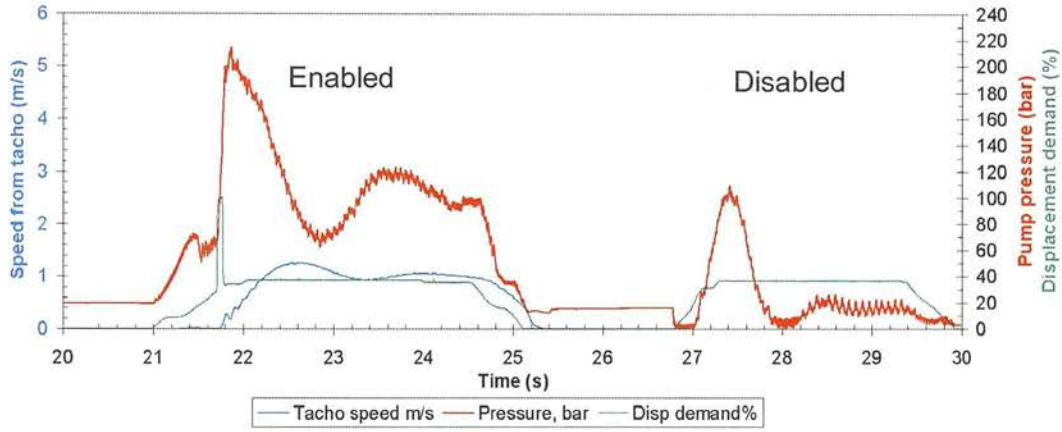


Figure 5-48: Test 6-2; 'jerk start' feature; red=pressure, blue=motor speed, green=pump displacement demand

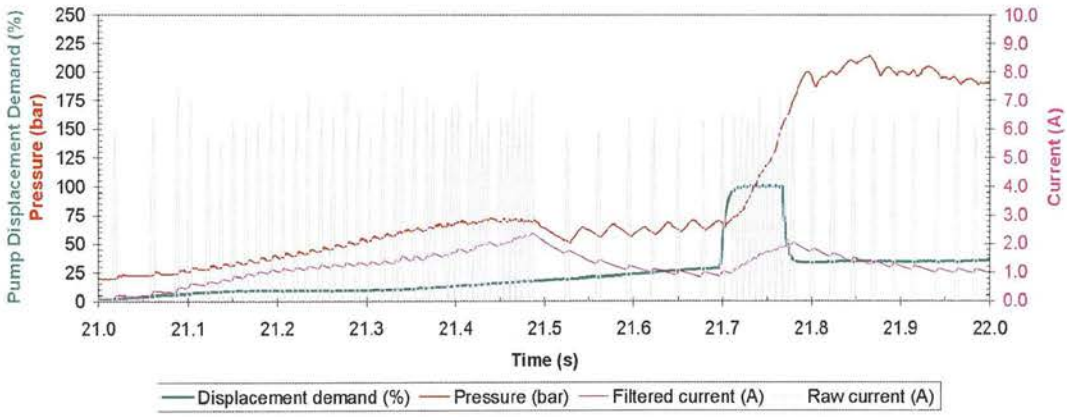


Figure 5-49: Detail of 'jerk start'; red=pressure, light pink=unfiltered current, pink=filtered current, green=displacement demand

The vehicle was placed on a hill and given a 20% speed demand with the joystick ('tight' mode was selected). At t=27, the response can be seen with the 'jerk start' feature disabled; the vehicle fails to start and all of the DDP flow leaks away in the motor. At t=22, the response is seen with the 'jerk start' feature enabled. The displacement demand steps up to

100% for ~100ms, enough to raise system pressure to 200 bar. The vehicle starts almost immediately and thereafter stabilises at a steady speed.

The current trace shows that a transition to full strokes occurred at t=21.5s; at t=21.72s the jerk start routine started as shown by the sudden increase in the demand signal and the resulting increase in pulses on the un-filtered current trace. At t=21.8s the frequency of the pulses in the current trace returned to the same level as before t=21.7s.

The ability of the DDP to step from 20% to 100% displacement and back again, in less than 100ms, is crucial to the application of this technique.

5.4.3 Position control mode

Slow speed, loose control

For test 7-1, the position handwheel was moved forwards and backwards to create steps of position demand. The vehicle was in 'slow' mode, and the 'loose' filter was selected to give smooth s-curves of position demand. The highest speed achieved in this test was 15% of the maximum speed of the vehicle.

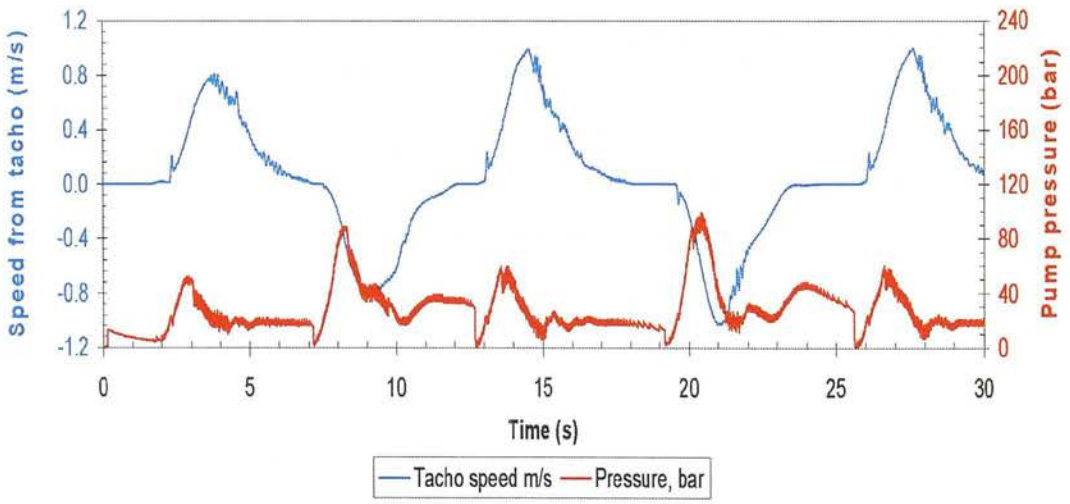


Figure 5-50: Test 7-1; blue=speed, red=pressure

It is notable that the pump pressure peak during the forward acceleration (speed=+ve) was lower than in the reverse movements; these peaks would be expected to be symmetrical. This could have been due to a slight gradient of the ground where this test took place.

The displacement demand signal was integrated in post-processing to calculate a theoretical position, assuming that the vehicle speed is linear with DDP displacement; this 'position from

flow demand' is compared below with the measured position to assess the accuracy of open-loop position control.

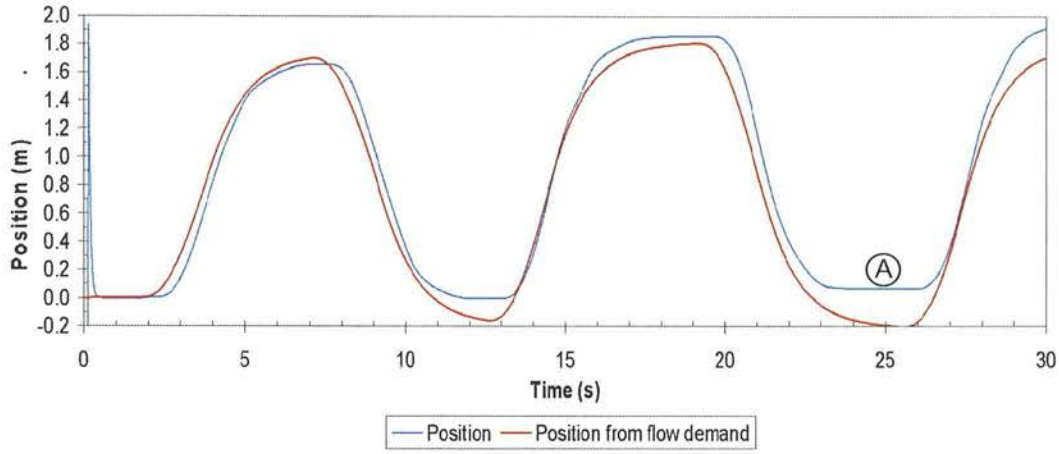


Figure 5-51: Test 7-1; actual and demand position

The actual position of the vehicle followed the demanded position fairly closely. The major deviation in form is seen near zero vehicle speed (A). Here the stiction in the drivetrain combined with the motor leakage to cause the vehicle to stop before pump flow had completely stopped.

'Fast' speed, 'loose' control

The open-loop position control was also investigated in the 'fast' speed control mode with the 'loose' filter active; here vehicle speed reaches 53% of maximum.

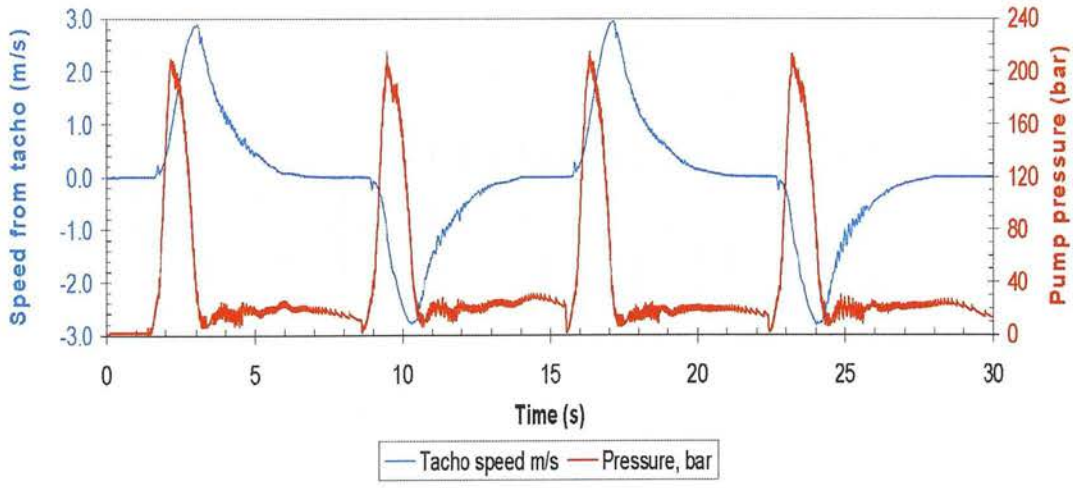


Figure 5-52: Test 7-4; blue=speed, red=pressure

The shape of the speed trace matches closely the form predicted from the simulation of the control algorithm shown in Figure 5-27.

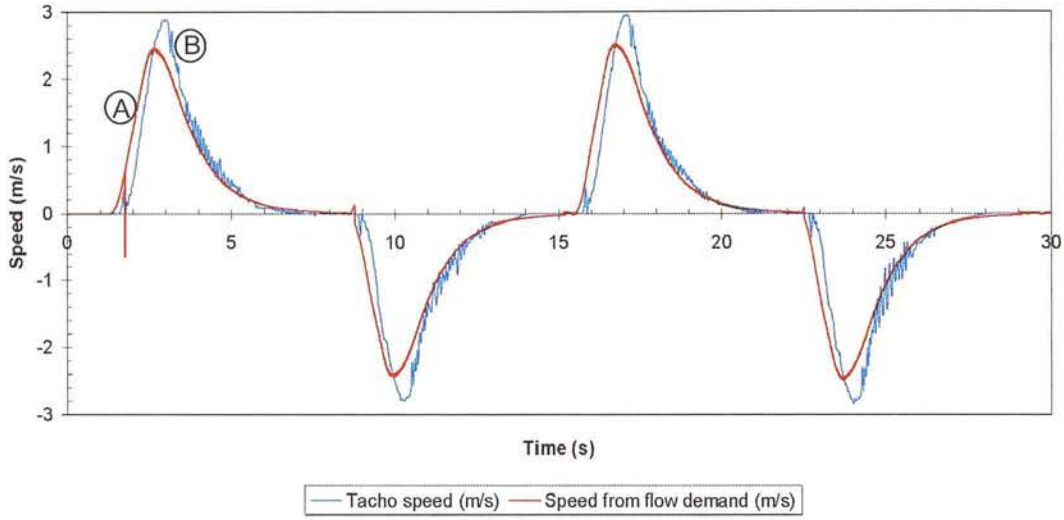


Figure 5-53: Test 7-4; actual and demand speed

The actual speed lags the speed demand due to the fluid which is stored in the compliance during acceleration, when the pressure is rising. Approximately 18.5cc must be displaced into the compliance to raise system pressure to 200 bar (arrived at by an experiment described in section 6.4.1). With full motor displacement, each metre of vehicle movement required 91.8cc be displaced into the motor (see Figure 6-8 p.303). The 18.5cc volume stored in the hydraulic compliance at maximum pressure, represents a distance of 0.201m.

At t=1.5s to t=2s (A), the pressure is rising; fluid from the DDP is stored in the compliance causing the actual speed to lag the speed demand. At t=2.5s to t=3.5s (B), the pressure is falling and the fluid stored in the compliance returns to the system causing the actual speed to catch up with the speed demand.

The theoretical position is compared below with the measured position.

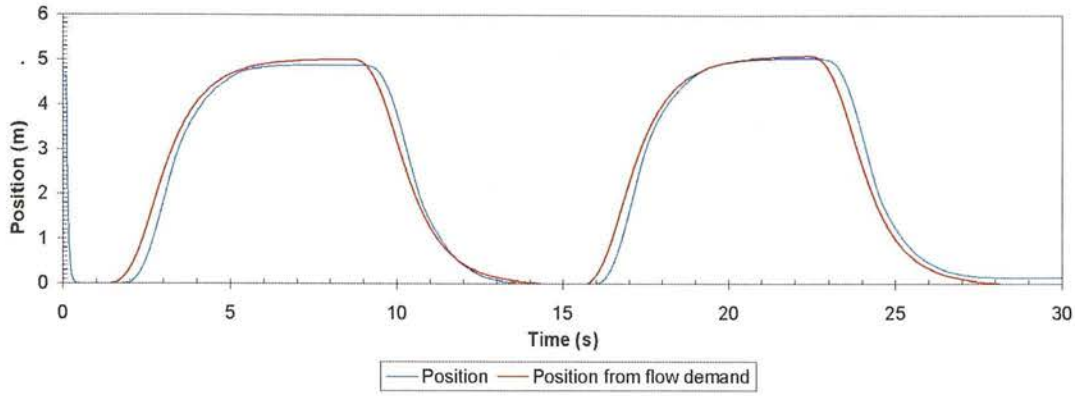


Figure 5-54: Test 7-4; blue=actual position, red=demanded position

The vehicle position followed the position demand more closely than in the low-speed test, because motor leakage and stiction is of less relative importance at higher speed.

The position sensor signal for this test was differentiated to calculate actual vehicle speed. This is compared below to the vehicle speed as measured by the tachogenerator on the motor shaft.

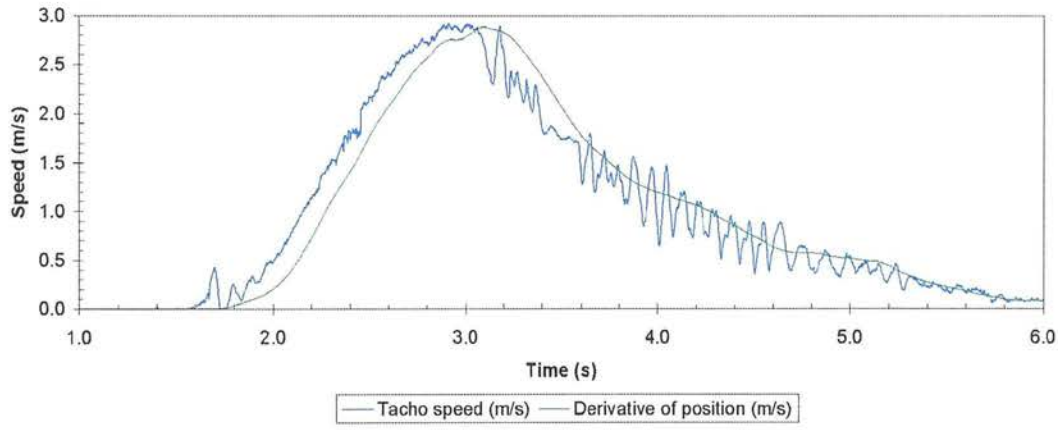


Figure 5-55: Test 7-4; blue= speed from motor tacho, green=derivative of measured position

The motor speed displays oscillation at approximately 13Hz during deceleration due to interaction between the overcentre valve and the driveline compliance, but this oscillation is absorbed completely in the driveline compliance; the resulting vehicle speed was smooth and there was no perceptible pulsation to the driver.

Fast speed, 'tight' control

For test 7-5, the position handwheel was moved in series of steps in the 'fast' speed range with the filter in 'tight' mode. This gave very fast and direct control of vehicle position.

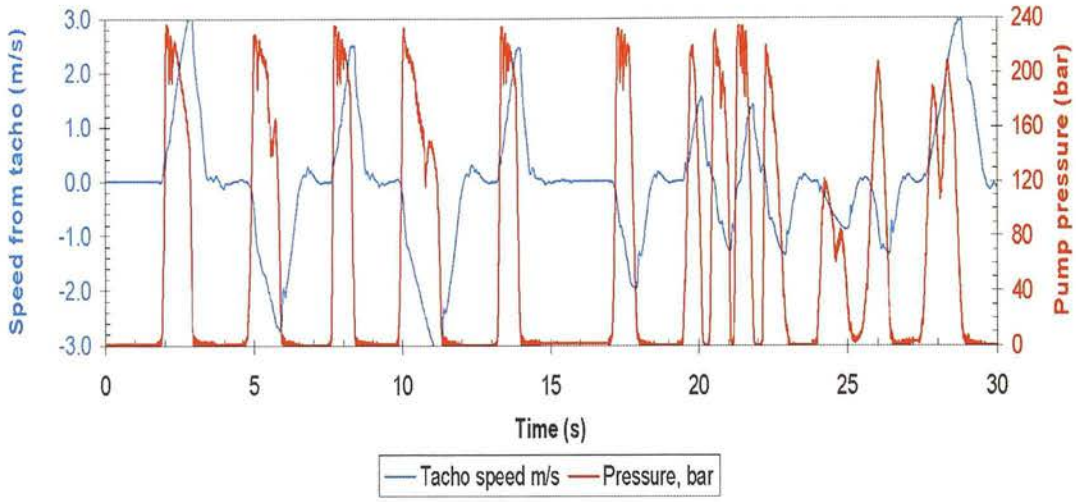


Figure 5-56: Test 7-5; blue=speed, red=pressure

The sharp position demand steps meant that system pressure reached the software pressure limit, causing the actual displacement of the pump to be momentarily less than the demanded displacement. The effect of this on the position control is shown below.

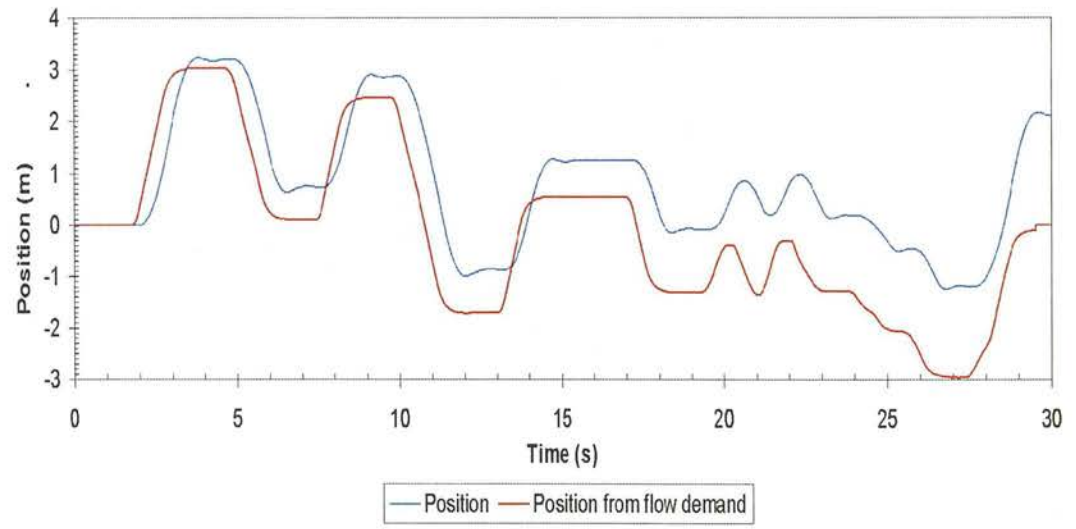


Figure 5-57: Test 7-5; blue=actual position, red= demanded position

The effect of the software pressure limit being activated is that an error accumulated between demanded and actual position. This is because the system controller always assumed that the displacement demand was followed, as it had had no way of updating its internal estimate of actual position due to pressure limiting in the DDP.

This behaviour would be improved if the system controller received a signal from the DDP controller of the actual displacement delivered; this would be easy to implement with a bi-directional communications bus (such as CANbus). Such a system would allow accounting of

every unit of fluid displaced into the load, even if the pressure-limit was frequently activated. This is possible with a conventional swashplate pump with a hydraulic pressure limiter, but only if a swash angle feedback sensor is fitted.

Software pressure-limiting with a DDP requires that a pressure sensor be fitted. The system cost could be reduced by limiting pressure only with a pressure-relief valve. This would not allow accounting of volume, and would cause large energy losses in aggressive driving, but may be acceptable for cost-sensitive applications with modest dynamic requirements.

5.4.4 Comparison of full stroke and part stroke modes

These tests investigated the effect of the DDP flow control algorithm on low-speed position-control creeping behaviour, responding to step demand of position of +/- 0.20m. Both tests involved a similar sequence of movements at very low speed; the vehicle speed did not exceed 7% of the maximum speed during the tests. The controller was in the 'slow' speed range with the 'loose' filter activated. These conditions made the effect of pulsation from the DDP more obvious than in other tests.

In test 9-6, the DDP delivered flow in 17% part strokes. In test 9-5 the DDP controller delivered flow in full strokes only. The results from these tests are shown below.

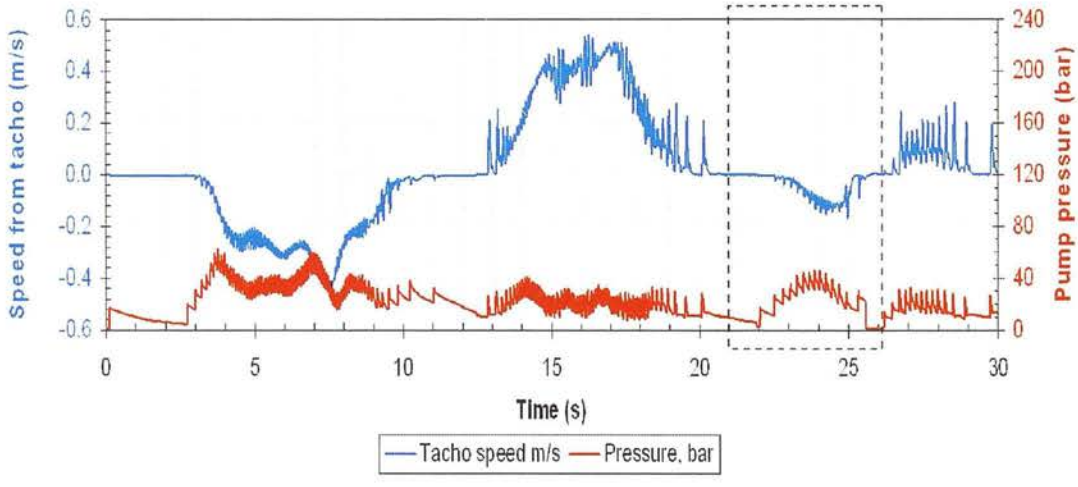


Figure 5-58: Test 9-5; creeping with full strokes; blue=speed from motor tacho, red= pressure

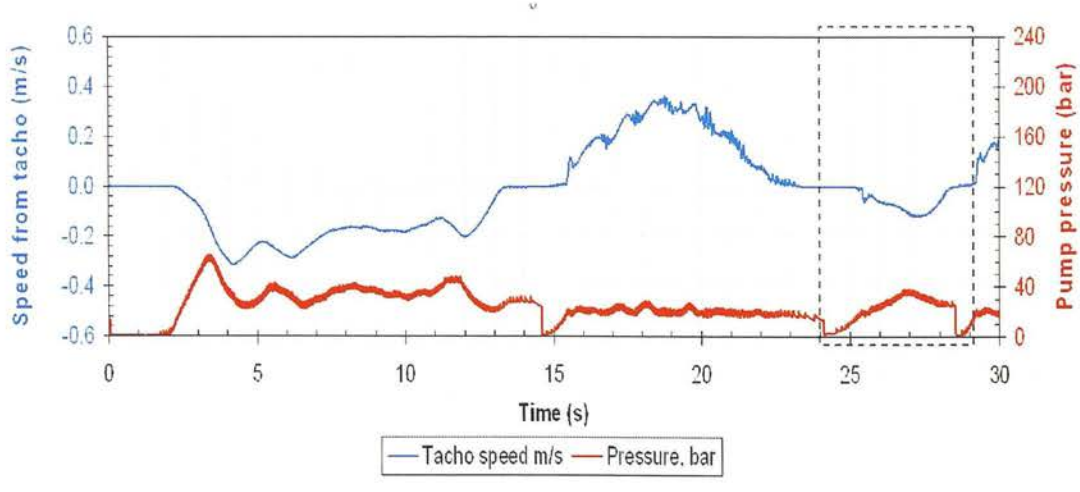


Figure 5-59: Test 9-6; creeping with part strokes; blue=speed from motor tacho, red=pressure

It is obvious that both pressure and motor speed were much smoother with part strokes than with full strokes. The area in the dashed boxes, where the vehicle moved very slowly (reaching only 2.2% of maximum vehicle speed), is examined in greater detail below.

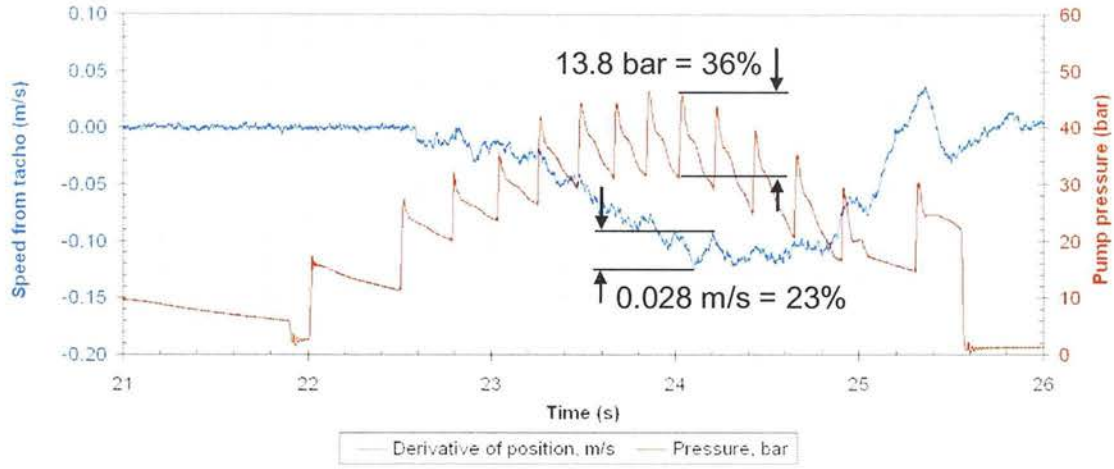


Figure 5-60: Test 9-5; Detail of pressure and speed pulsation when creeping with full strokes; blue=speed from differentiating the position sensor, red=pressure

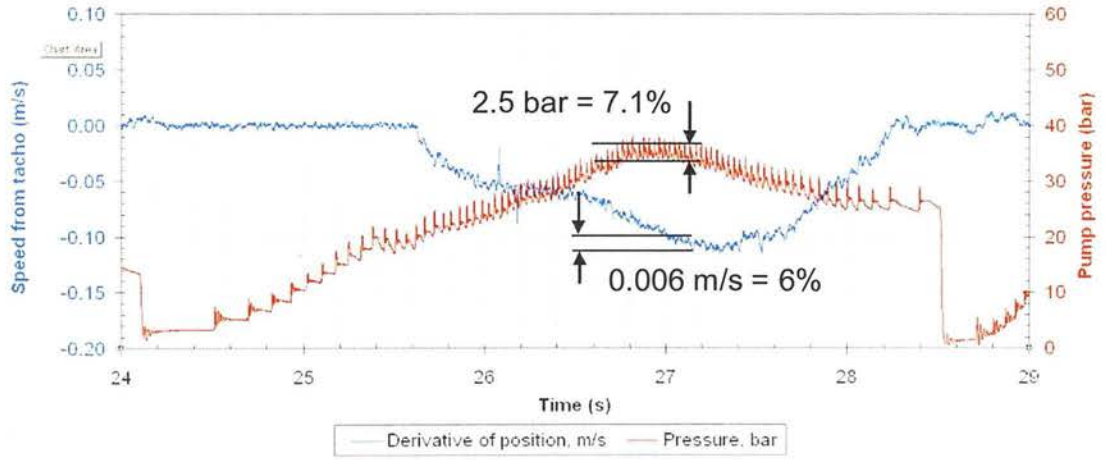


Figure 5-61: Test 9-6; Detail of pressure and speed pulsation when creeping with part strokes; blue=speed from differentiating the position sensor, red=pressure

In the figures above, vehicle speed is calculated by differentiating the vehicle position trace; this measures the actual speed of the vehicle as filtered by the drivetrain compliance.

The pressure pulsation with the part stroke mode was 36% (peak-peak); this reduced to 7.1% in the part stroke mode, slightly more than might be expected from the 1:6 ratio of the magnitude of the volume displaced by each pulse. In the full stroke mode, there was clear speed pulsation in the full stroke mode of 23% (peak-peak), which was perceptible to the

driver; the vehicle was felt to move in discrete steps. In the part stroke mode, the speed pulsation was measured as 6%; this is little more than the noise on the signal when the vehicle was stationary, and the vehicle motion was perceived to be completely smooth.

Examination of the acceleration and current traces for the same very low speed portion, allows correlation between vehicle acceleration pulses and each LPV actuation event:

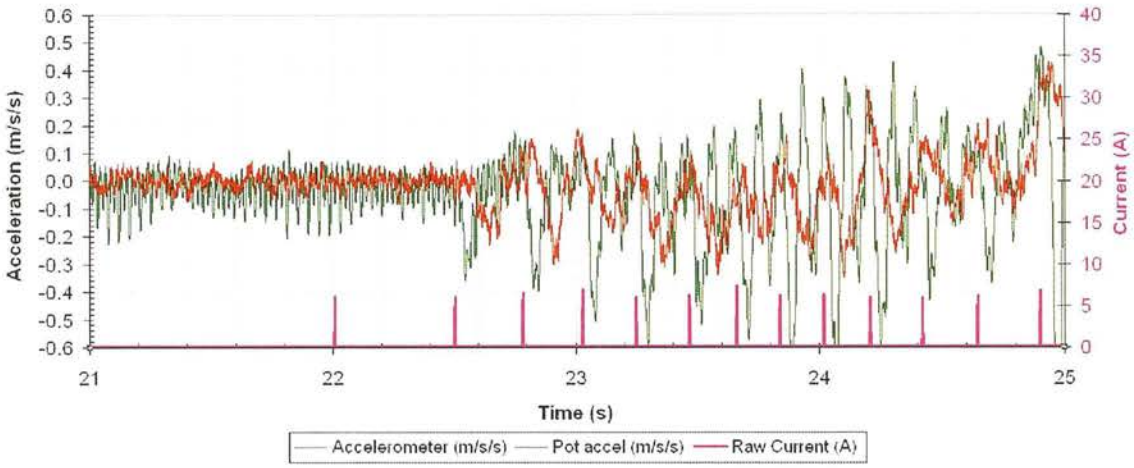


Figure 5-62: Test 9-5; full stroke mode; red=acceleration from double differentiation of measured position, green=filtered accelerometer signal, pink=unfiltered current.

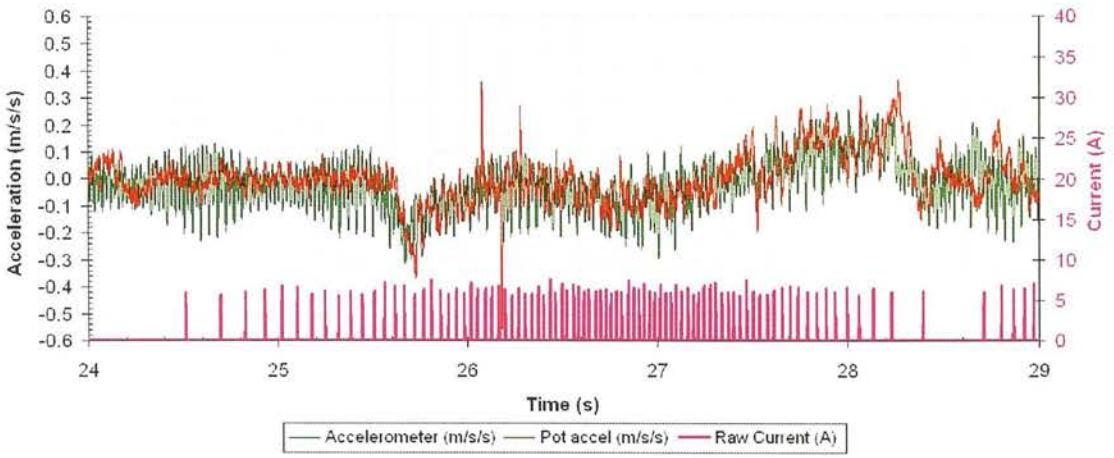


Figure 5-63: Test 9-6; part stroke mode; red=acceleration from double differentiation of measured position, pink=unfiltered current, green=filtered accelerometer signal

For the above two figures, acceleration was measured in two ways. The green trace shows the accelerometer signal, filtered with a 40ms moving-average to remove the chassis vibration caused by the engine; without this filter, the engine vibration swamped the acceleration pulses caused by the DDP. The red trace shows the result of double-differentiating the vehicle position sensor.

Each current pulse indicates when a LPV was actuated. In the part stroke mode the frequency of these pulses was 6 times larger than in the full stroke mode, and the volume delivered in each stroke was 6 times smaller.

In the full stroke mode, the current pulses can be clearly correlated with acceleration pulses, indicating that the motor was rotating in discrete steps in response to each full stroke from the DDP, as was perceived by the driver. By contrast the part stroke mode showed little correlation between acceleration pulses and current pulses, and the magnitude of the acceleration pulses was approximately 6 times less; it is difficult to discern the pulses due to the DDP, from the noise. There was no perceptible pulsation to the driver in the part stroke mode.

Overall, the results from tests 9-5 and 9-6 demonstrate that it is possible to obtain completely smooth creeping behaviour with a DDP transmission, as long as part strokes are used.

5.5 Conclusions

The questions posed at the beginning of this chapter have been answered. In particular:

- The DDP has been applied to a vehicle propel transmission with a switched-displacement axial piston motor, in a pump-controlled open circuit configuration, with an overcentre valve for hydraulic braking.
- A propel system controller has been described which allows smooth open-loop control of vehicle acceleration, speed and position.
- The fast response of the DDP has been shown to allow the coordination of DDP displacement with the switching of motor displacement such that the vehicle motion is continuous, despite the sudden change in pump flow and pressure.
- Similarly, the switching of the digital directional valve can be coordinated with the DDP displacement such that continuous motion through the zero-speed region is possible.
- Detailed experimental data of the effect of DDP pulsation on pressure and vehicle acceleration has shown that the part stroke mode of DDP displacement is crucial to achieving smooth motion control at low creeping speed.
- The DDP has been shown to exhibit significant control advantages to the swashplate pump at very low creeping speed.

6 Simulation of the DDP propel system

6.1 Introduction

The DDP produces a higher level of flow pulsation than conventional variable-stroke piston pumps. To investigate whether this pulsation is acceptable in vehicle applications, it is desirable to create a time-domain computer model of the effects of the DDP flow pulsation on an overall vehicle system. This chapter describes such a model of the the system described in the previous chapter and compares the results from the model with experiments.

Rampen's first software model of DDP behaviour was written in the 'Basic' language. Although the fundamentals of the flow algorithm matched experimental data well, the ability to model systems was limited by this approach. Md. Ehsan (Ehsan 1997) created a more complex DDP model in Simulink for time-domain simulation of a free-piston engine with a DDPM serving as a "hydraulic crank". The thermodynamics in the engine were complex, but the hydraulic system rather simple. Ehsan's virtual DDP was not fundamentally a physical simulation of the internals of a DDP, rather a "function generator" which produced patterns of half-sinusoids, phased relative to the shaft position, and selected by logical switches. These ideal half-sinusoid signals were then treated as positive or negative flow signals which entered a linear fluid compliance, connected to the working chamber of a heat engine.

This "function generator" approach has major benefits in terms of reducing simulation time, but has limited accuracy due to a number of real-world effects:

1. The dynamics of the HPV cause reduced effective displacement at high speeds, because the spring and mass of the HP poppet. As the speed of the machine increases, a point is reached whereby the HPV remains open some time after TDC, allowing a fraction of the volume generated by the machine to flow back into the cylinder. This has the effect of reducing the volume displaced at high speeds, as was observed in the SD1B machine above 2500rpm (Figure 2-14, p.34) This is a well-known issue for check-ball pumps in general. (Edge & Brett 1990, Johnston 1991)
2. Flow forces on the LPVs cause the closing time to be a function of the phase angle at which it is actuated (particularly important in the 'variable part stroke' mode)
3. Finite compressibility inside the cylinder causes a reduction in the effective displacement as pressure increases.

4. When each cylinder is pressurised, leakage occurs from the cylinder at the piston-cylinder, cylinder/top end bearing and piston pad bearings.

5. Deformation of the HPV in the seat area and the finite mass of the HPV poppet can cause a pressure over-shoot within the cylinder before the HPV opens – this is effect is strongly affected by the amount of working chamber dead volume. High-frequency pressure oscillations may occur as a result of this overshoot (Johnston 1991, Nguyen-Shaefer 1998).

M. Eshraghi (Lynn et al. 2005) created with the author's support a fairly detailed model of the dynamics inside a single cylinder of a DDP, using the Amesim software tool (Imagine S.A.), which was similar in principle to the model of a check-valve pump described by Johnston (1991). The equations of motion of the LPV were modelled in a one-dimensional lumped-parameter form, the damping of each sliding interface being modelled separately. The solenoid current was generated by a one-dimensional simulation of the electrical circuit, linked to the LPV motion with an ideal solenoid element. The cylinder itself had dead volume, compliance and three leakage sources. The HPV was modelled in a one-dimensional lumped parameter form without flow forces. Both HPV and LPV had elastic end-stops.

This model was capable of exhibiting all of the non-ideal effects listed above except for no. 2, which would have required detailed experimental data of flow forces, which was not available. However, it had some important drawbacks:

- The model took 20 minutes to simulate few seconds of pumping work in a single cylinder with a 1GHz PC, due to the extremely short time constants involved. This severely limited its application to the problem considered herein, a simulation of vehicle motion over many seconds, including a 6-cylinder DDP.
- Matching the real machine's non-ideal behaviour required the acquisition of dozens of parameters from either experimental or FE/CFD work.
- Simulating the real actuation delay and transit time of the valve meant that the control code in the simulation had to include all the details of the valve timing algorithm used in the real embedded controller.

This experience led the author to reject detailed physical modelling as the basis of a DDP model for vehicle system simulation.

At the outset, the purpose of the model must be considered. It is always possible to find more detail to analyse, but a fundamental constraint exists: the more complex the model, the shorter the time constants of the dynamics involved, and the longer the simulation will take. If the objective is simulation the behaviour of a vehicle over a driving cycle, then everything that

happens within the DDP can be considered to be inside a black box. It is what comes out that matters, rather than what happens inside.

For the purposes of system simulation, the DDP can be reduced to Ehsan's 'function generator' if the effects can be neglected of HPV dynamics, finite LPV actuation time and chamber compressibility.

The results in Section 2 (Figure 2-14, p.34) of the effective displacement vs. speed for the SD1B machine, similar to the DDP used in the demonstrator vehicle, show that at 1440rpm (the nominal speed of the DDP shaft in the demonstrator vehicle), the effect of valve dynamics on the amount of flow delivered by each pumping stroke of the DDP is very small. At 5 bar, 1440rpm, the SD1B delivered 34.5 cc/rev, very close to the geomterical displacement of 34.9cc/rev.

The fact that the LPV has significant delay and transit time can be ignored if it is assumed that these times are consistent, and that the controller has an accurate compensation for them. The results in 5.3.1 show that this is achievable.

However, the effect of compressibility on the volume delivered by each stroke cannot be ignored due to the relatively large dead volume of the DDP. The results of the SD1B (Figure 2-14, p.34) show that at 200 bar, the DDP delivered only 88% of the flow delivered at 5 bar.

As a result of these considerations, the detailed physical modelling approach was rejected, and a DDP model was created which was similar to Ehsan's 'function generator', but with a modification for the effect of compressibility.

At the outset a decision had to be made on what parts of the vehicle control system were important to simulate. Initially it was thought desirable to reproduce all of the system controller signal flow and logic functions, and simulate the synthesis of the displacement demand signal from the operator inputs. The actual recorded operator control inputs from the RS232 data could then be fed into the Dymola model, which would simulate the action of the system controller and produce the DDP displacement demand signal and the external solenoid valve signals. This approach was rejected for the following reasons:

- The operator demand signals were only available at 11Hz update rate from the RS232 data, and this was though to be too coarse for an accurate simulation.
- The actions of the system controller are primarily feed-forward and therefore deterministic.
- The effort involved in transferring all of the system control code into the Dymola model would have been considerable.

A more direct approach was to stimulate the model not with the recorded operator demands, but with the recorded displacement demand signal sent to the DDP by the system controller. The operation of the system controller was not the primary focus of the simulation, rather it was the detailed dynamics of the hydraulic system when excited by the output of the DDP.

6.2 Literature review

6.2.1 The linear approach

Some have successfully applied classical linear analysis to the problem of hydrostatic transmissions (e.g. Krus 1991, Rameda et al. 2003). Carefully applied, this method can yield useful insights into the frequency response and control stability around an operating point. Lennevi and Palmberg (1995) used linear analysis to study the influence of engine compliance on hydrostatic transmission performance. In their analysis, the pressure in the system was proportional to the integral of the difference between the flow produced by the pump and that consumed by the motor i.e. the fluid connection between pump and motor was characterised as a linear compliance. A 2^{nd} order linear transfer function was derived from this relationship and studied in the frequency domain, yielding insight into the effect of the engine governor behaviour on the overall vehicle performance when subject to small-signal perturbations.

However linear analysis cannot model the important non-linearities common to all hydrostatic transmissions of (amongst others) pressure-limiting, non-linearities in pump displacement controls and stiction in tyres (Tilley & Tomlinson 1991, Hayashi 2001). In addition to these features, the golf buggy vehicle studied here contains an overcentre valve, which has the fundamentally non-linear characteristic of "cracking" when the force balance on the poppet reverses.

However the major barrier to linear analysis of a DDP hydrostatic transmission is the discontinuous flow pulses produced by the DDP itself. Even if the Fourier transformation of the discontinuous flow waveform could be derived, the pulses could interact with the system non-linearities to create complex time-domain behaviour.

For the system studied here, it was found experimentally (Section 5.4.2) that, even ignoring the non-linearity of the DDP pulses, the mechanical/hydraulic system does not exhibit the classic linear system characteristic of *linear superposition*. The effective damping factor of the response of vehicle speed to pump flow steps was observed to change over the full operating speed of the vehicle.

For these reasons, linear analysis was thought unable to reproduce the behaviour of the system studied here, and a non-linear time-domain simulation approach was taken.

6.2.2 Non-linear time-domain simulation

Åström et al. (1998) give a historical overview of methods of simulating physical systems. Until the advent of the digital computer in the 1960's, analogue computing methods dominated, whereby signals were represented as voltages and systems were assembled from operational amplifier circuits functioning as integrators, summing junctions and multipliers. Parameters were controlled by potentiometers. Although tedious to set up and lacking absolute accuracy, it would be the 1980's before digital computers could match them for speed, and even today the interactive learning experience of exploring the behaviour of a dynamic system by changing parameters, while the model runs in real time, is not matched by many software tools.

Hydraulic systems present a distinctive challenge to digital computer simulation, because such systems are usually numerically "stiff"; there is a wide range of time-constants, some of which become dominant only at discontinuous events such as end-stops. This requires a variable-step solver and places extreme demands on the numerical integration method (Richard et al. 1990).

In the 1980's, block-based simulation modellers with graphical interfaces such as Easy5 and MATRIXx System Build appeared. Undoubtedly the most successful of these has been Matlab Simulink, which since it appeared in 1991 has became the de-facto standard tool for simulating dynamic systems and their controllers.

The process of implementing a Simulink model is similar to that of creating an analogue computer model, consisting of linking together the inputs and outputs of pre-defined blocks. A rich variety of linear and non-linear blocks is provided, and a selection of integrators with which to solve them, some of which are suitable for stiff models. The program particularly excels at simulating digital control of dynamic systems, allowing the mixture of non-linear continuous models, state machines, sampled data systems and C-code embedded software, in one simulation.

Just like the analogue models, signals only flow one way in Simulink. This is quite unlike a typical hydraulic schematic, where a connection between a valve and pump and a cylinder is bi-directional. If the pump speeds up, the cylinder speeds up; if the cylinder encounters an increased load the resulting pressure rise will be rapidly felt by the pump, typically causing a reduction in speed of the prime mover (Lennevi & Palmberg, 1995). The modelling of hydraulic circuits in Simulink involves the explicit declaration of all such feedback loops

which are implicit within the schematic. The model representation differs dramatically from the schematic, making verification and maintenance of the model more difficult. Simulink has been widely used for hydraulic system modelling, possibly because it is familiar tool which is widely available, but despite this, it is rather ill-suited to the domain.

BathFP (Richard et al. 1990) was developed at Bath University specifically for engineers to use for simulation of coupled hydraulic and mechanical systems. This was one of the first packages to allow a system to be described by simply drawing a schematic, the feedback loops implicit within hydraulic connections being hidden from the user. Each component could then be associated with a particular model, allowing the user to select the complexity of model appropriate for each component depending on its importance within the overall circuit.

Amesim (IMAGINE S. A.) developed this idea into the arena of full multi-domain modelling, allowing systems to be constructed linking the control, electrical, mechanical and hydraulic domains. Each domain has its own component library and connection types, with interface blocks to other domains (e.g. a solenoid links the electrical and mechanical domains via the magnetic domain).

Such programs offer a major advance in terms of usability for the vast majority of common tasks covered by the built-in library of models. However the construction of a new model for a new component (such as a DDP) is a difficult task which requires detailed understanding of the underlying simulation code. The underlying code of Amesim models is inaccessible to the end user. Amesim attempts to tackle this problem by supplying a large number of primitives in its Hydraulic Component Design library. However, Amesim lacks the discrete digital control elements required to model the interrupts in the DDP embedded controller, synchronised to the shaft position.

Dymola (Dynasim AB) is a user interface and solver for models based on the object-oriented Modelica language. The use of Differential Algebraic Equations (DAEs) instead of Ordinary Differential Equations (ODEs) allows the construction of classes with intrinsic bi-directional signal flow. The separate physical domains of electrical circuits, magnetic circuits, mechanics and hydraulics can be modelled in one simulation, while being controlled by state machines, continuous, sampled, or interrupt-driven controllers. Hydraulic actuators can be coupled to a 3D kinematics model, allowing the accurate simulation of mobile hydraulic machines such as excavators (Beater & Otter 2003).

The combination of pre-built hydraulic and mechanical libraries and well-developed discrete control blocks makes Dymola well-suited to simulations of mobile machines including a DDP. Particularly important for hydraulics, is Dymola's powerful handling of instantaneous

events; this allows the computationally-efficient handling of the discontinuities which occur during cavitation, endstop or pressure relief situations, as well as within the DDP itself.

For these reasons, Dymola was selected as the tool for simulating the DDP transmission system.

6.3 Implementation of the hydraulic system model

Components from the HyLib hydraulics library (Beater 2000) were used as building blocks to speed model development. The HyLib Extension (Claytex Ltd.) allows the use of engineering units such as "cc" and "bar" in place of unwieldy SI units such as "m^3/radian" and "Pascal".

The top-level system diagram from the Dymola environment is shown below:

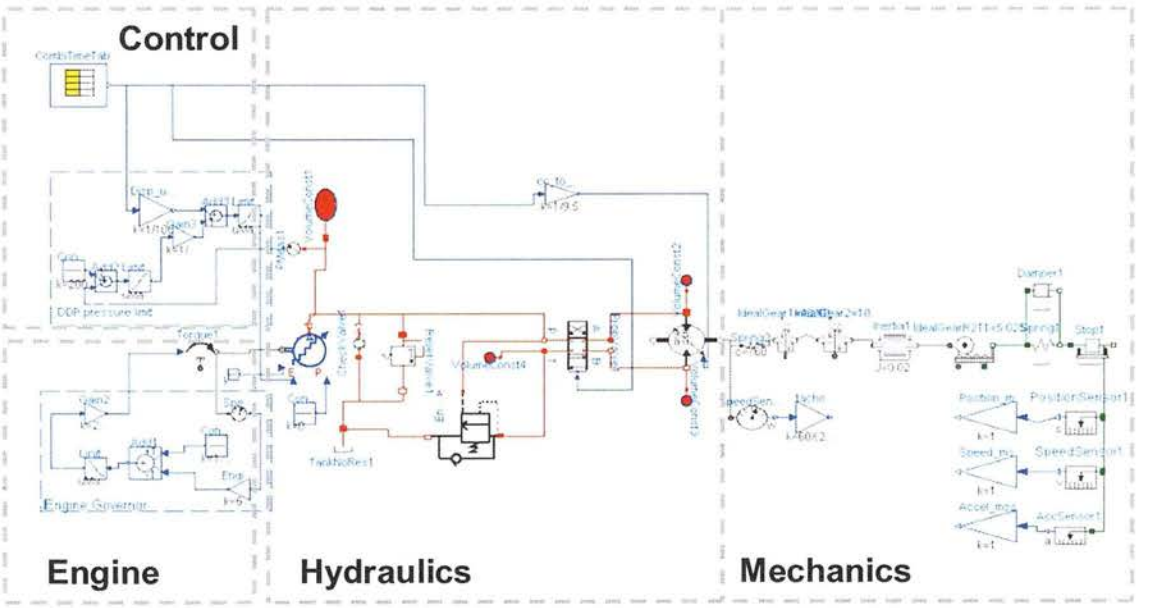


Figure 6-1: Top level of the Dymola model

The sections labelled above are considered in more detail below.

6.3.1 The Control section

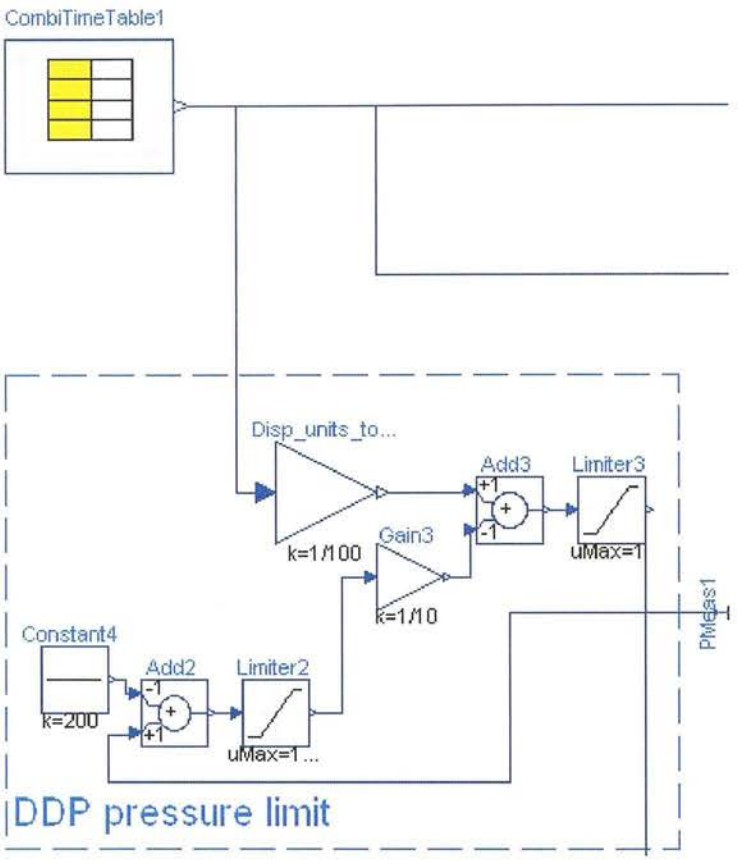


Figure 6-2: The Control section of the model

The Control section has three major functions: import of stimulating control signals from the experimental results, import of measured data of the actual system response for comparison with the model, and implementing the DDP pressure-limiting function.

The *CombiTimeTable* block imports the experimental data into the Dymola workspace. This consists of 25 times series, each collected at 1kHz sample rate. Of these, three were used to stimulate the model:

- the displacement demand sent by the system controller to the DDP controller, converted from a voltage to a displacement demand (0 to 100%)
- the command to the directional valve (forward or reverse);
- the motor displacement demand (full or reduced).

The rest of the signals were measured data which were used to compare the response of the model with the response of the real system. The most important of these are:

- measured pump pressure, from the pressure transducer;

- measured motor speed, from the tachogenerator;
- engine rpm, measured from the frequency of DDP shaft trigger pulses sampled by the DAC;
- position of the vehicle, from the "yo-yo" potentiometer;
- acceleration of the chassis, from the accelerometer;
- instantaneous current supplied to the DDP FET switches, from the current clamp.

This section also includes the simulation of the DDP pressure limiter control function, built into the DDP controller. This is modelled as a linear reduction of simulated DDP displacement from the command displacement above 200 bar, such that by 210 bar the DDP is commanded to zero displacement. In fact the pressure limiter function in the DDP embedded software is more complex than this, but this simplification was found to give a reasonable approximation.

6.3.2 The Engine section

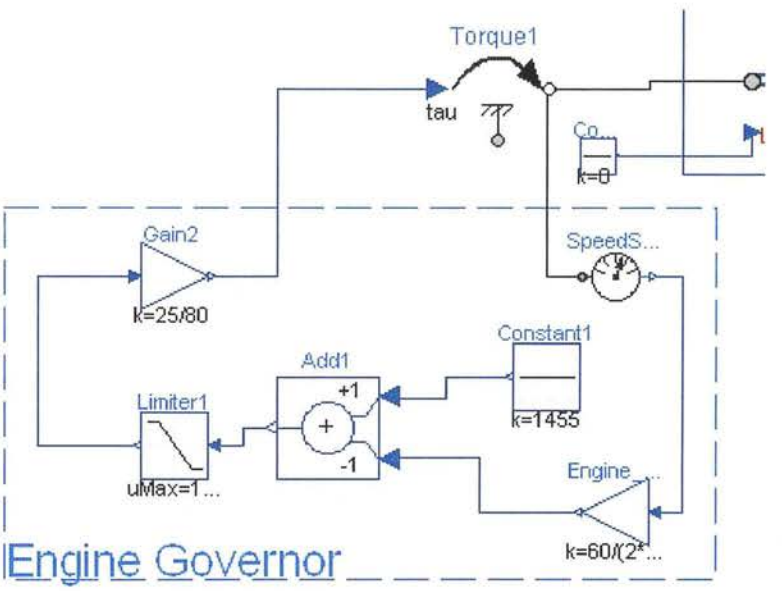


Figure 6-3: The Engine section of the model

The Engine section models the Honda GX340 petrol engine as an ideal torque source, the simulated governor acting to create a torque proportional to the instantaneous droop below the nominal setpoint of 1455rpm; the actual engine ran at twice this speed but was connected to the DDP via a 2:1 gear reduction. Above this nominal speed, the engine was assumed to provide zero torque – there was no need to model engine braking, as the DDP is incapable of

providing an accelerating torque to the engine. The engine inertia was lumped into the rigidly-connected DDP model.

6.3.3 The Hydraulics section

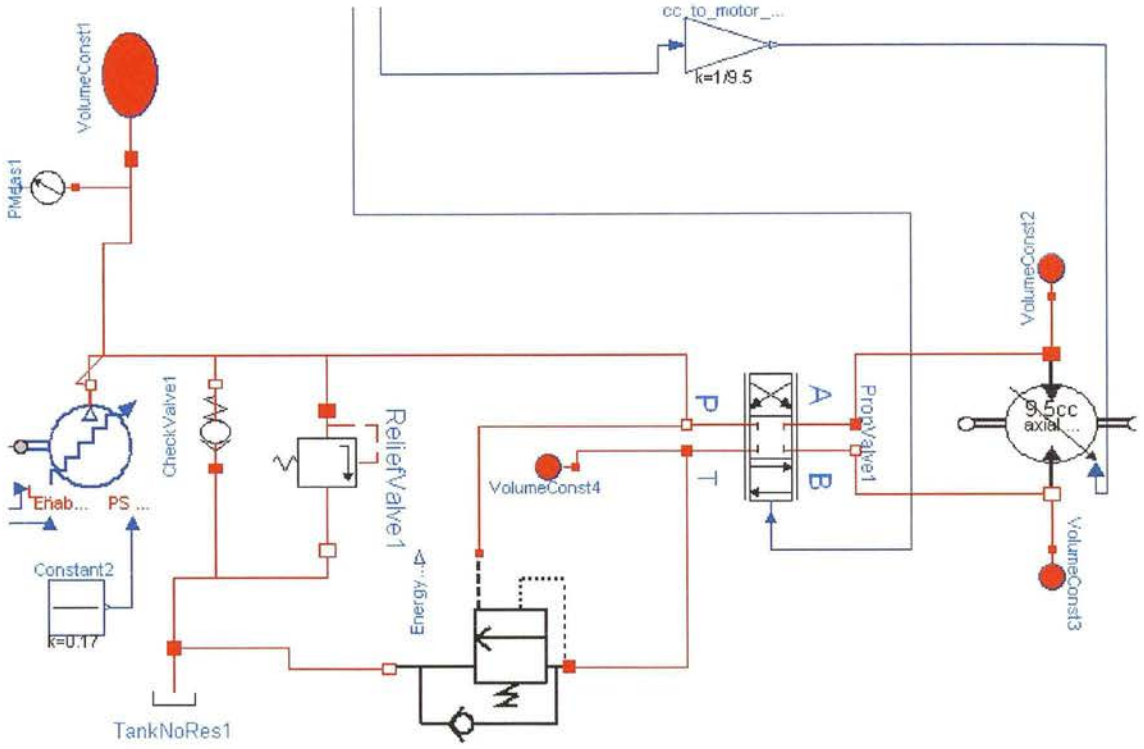


Figure 6-4: The Hydraulics section of the model

The Hydraulics section models the hydraulic system dynamics. The red lines in the diagram above are HyLib ideal hydraulics connectors; these constrain the pressure at each connected node to be identical, and apply a flow continuity equation such that the flows at each node sum to zero.

The DDP model is dealt with in its own section below. For initial development of the model, the DDP was approximated to an ideal flow source, producing flow as the product of shaft speed, maximum displacement and the displacement demand fraction.

The system characterisation tests showed that the hydraulic compliance could be modelled as a linear pressure compliance equivalent to a dead volume of 1.7 litres of oil with a bulk modulus of 16000 bar. The compliances from bulk modulus of the oil, hose expansion and strain of the components were lumped together into a single compliant volume. The smaller volume models at the ports of the motor and the overcentre valve are necessary for numerical stability and are a negligible 5cc in each case.

The system relief valve was modelled as being completely closed below 220 bar, with a linear pressure drop vs. flow rate characteristic above this pressure, of 1 l/min per bar above the cracking pressure (taken from the manufacturer's data). It should be noted that during normal operation, the DDP controller limited pressure by reducing displacement above a setpoint of 200bar; experimental and modelling results showed that the cracking pressure was rarely reached, so the valve has little effect on the system dynamics.

The check valve component in the model was inserted to prevent cavitation in the simulation. In the real system, this function is provided by the HPVs in the DDP; these act as check valves to prevent the DDP outlet pressure falling significantly below the tank pressure.

The directional control valve was modelled using the HyLib proportional valve model. In the simulation it was excited with digital demand signals only, so functioned as a four-port, two-position directional valve. A 20ms time constant modelled the actuation speed to be in approximate agreement with the data sheet for the Integrated 12C515M. A linear pressure drop of 25 bar at 20 l/min was introduced to match experimental data; all of the pressure drops between pump and motor are lumped together here.

The Lucas PM125 9.5cc axial piston motor is modelled using the HyLib *VarMot* block. As supplied, this models a variable displacement motor with inertia, internal and external linear leakage, and viscous speed-related losses. To this the author added pressure-dependent Coulomb friction to better model starting behaviour. The Coulomb friction parameter was chosen such as to give the motor 90% torque efficiency when starting at full displacement and full pressure. The internal leakage (i.e. A to B port) was assumed to be zero, while the external leakage to tank was chosen to match experimental data from the observed pressure decay of the system compliance under stalled conditions; all of the observed leakage was lumped together here. The calculated inertia of the rigidly-connected belt-drive pulley was added to the data-sheet value to give a lumped inertia for this motor

The HyLib library does not supply an overcentre valve model, so one had to be created. The detail of this submodel is shown below:

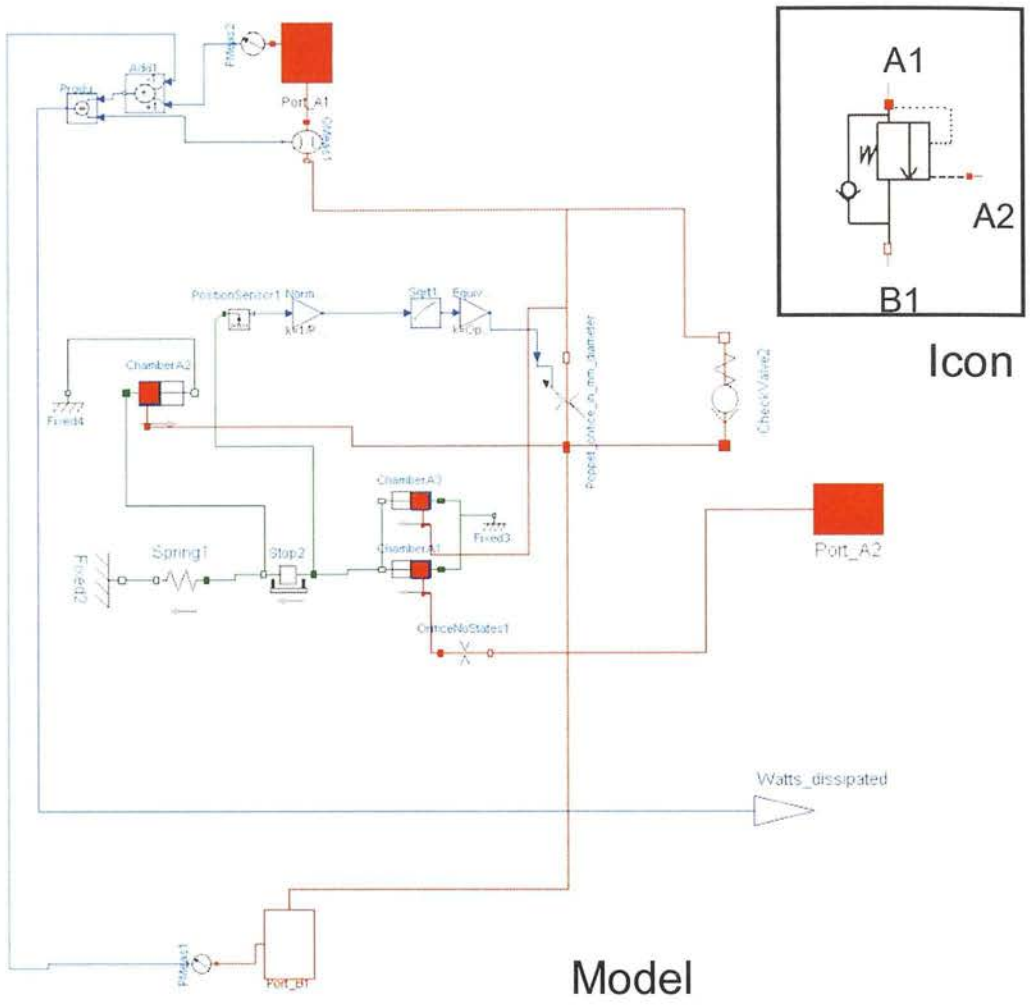


Figure 6-5: The overcentre valve model

The analysis of the force balance on the poppet by Andersen et al. (2005) was taken as a starting point, and a model built from Dymola blocks from the mechanical, signal and HyLib libraries.

The valve poppet was modelled as a mass sliding between end-stops with damping. The mass of 20g was calculated from inspection of the valve section drawing, while the damping constant of 20 Nsm^{-1} was chosen to prevent extremely high-frequency chattering behaviour in the simulation. In fact the dynamics of the simulated system behaviour were insensitive to this parameter at all sensible levels; it was observed that very high damping on the poppet increased system instability.

A spring acts to keep the poppet on its seat. This spring is opposed by the combined forces produced across the frontal area of the poppet and the much larger area of the pilot piston. The ratio of pilot area to frontal area was 8:1 for this valve. The pilot piston is fed through an ideal orifice of 0.8mm diameter (from measurement of the real valve). The pressure drop

across the poppet was modelled as an ideal turbulent orifice, the area of which was assumed to be proportional to the poppet position. At the fully-open position the diameter of the modelled orifice is 7mm, which matched the pressure vs. flow characteristic curve supplied by the manufacturer (Integrated ICE90).

Rather than an ideal turbulent orifice, the orifice models used by Beater in the HyLib library (Beater 2005) have both laminar and turbulent pressure drops for reasons of numerical stability, but the laminar pressure drop is only significant at very low flow rates (Re<10).

6.3.4 The Mechanics section

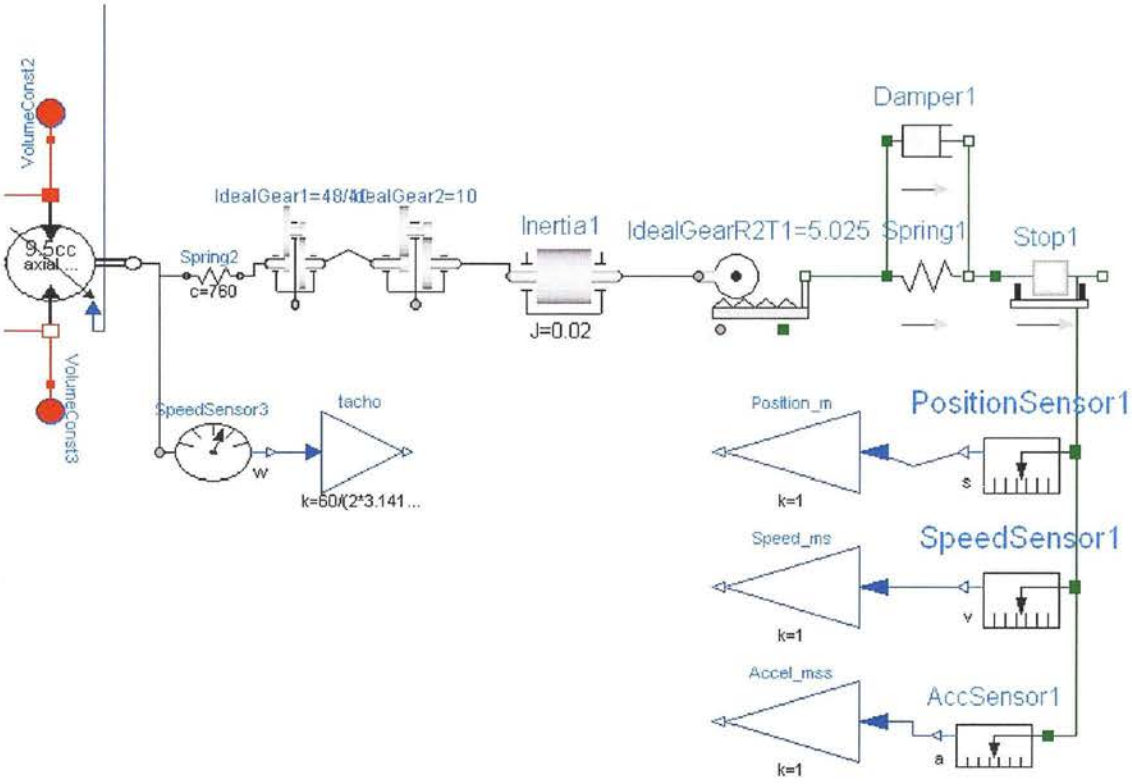


Figure 6-6: Mechanics section

The drivetrain of the vehicle was modelled as a coupled 1D rotational-translational system. The motor model included the lumped inertia of the drive pulley. The drive belt stiffness was modelled as a rotational spring, the stiffness of which was measured by applying a known torque to the drive pulley and observing angular deflection. The *Ideal gear* stages reflect the pulley ratios (48:40) and the differential final drive ratio (10:1). The inertia of the wheel was modelled in the *Inertia1* block – this was derived from the solid model. The transformation from rotational to translational domains was implemented in the *IdealGearR2T1* model as a rolling radius of 199mm (=5.025rad/m), from measurement of the vehicle wheel.

The tyre torsional stiffness was modelled in the translational domain as a spring of 102kN/m – this value was arrived at by measuring the angular deflection of the wheel when full tractive effort was applied, with the vehicle pushing against a stationary object. The value of the damper in parallel was tuned by comparing experimental traces of tacho velocity with the simulation results, to arrive at approximately the same damping factor as observed in experiment.

The *Stop1* block modelled the mass of the vehicle and rolling resistance. The vehicle was weighed at 360kg and an extra mass added of 90kg to account for driver and data acquisition equipment. The low-pressure tyres of a golf buggy have significant rolling resistance. This was estimated by placing the transmission in neutral (by disengaging the differential) and applying a force to the chassis through a spring balance until the vehicle started to move; this force was approximately 200N at break-away, falling to around 100N once the vehicle was moving. This force was split into two equal parts: a Coulomb part of 100N, constant with respect to speed and an additional Stribeck part, which starts at 100N at $v=0$ and decays exponentially as speed increases, as shown below.

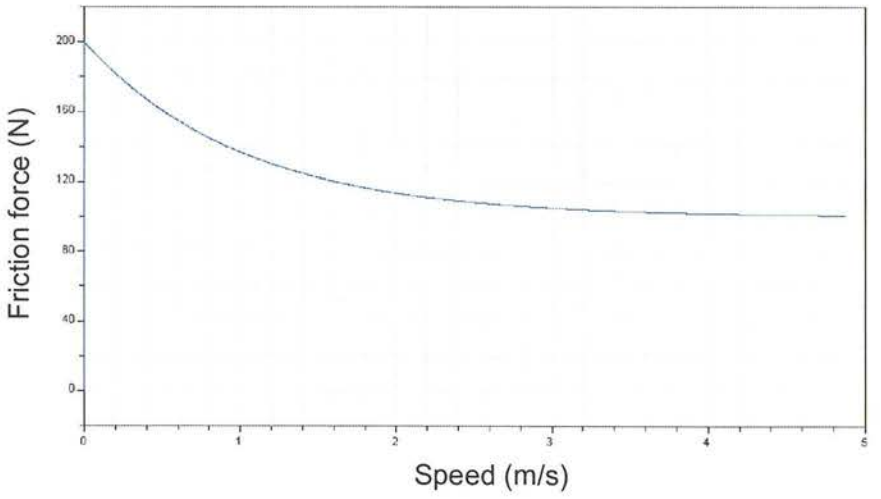


Figure 6-7 The simulated rolling resistance of the vehicle

When considering the first natural frequency of the system, it is useful to assume the drivetrain is perfectly stiff, and relate fluid flow and pressure at the motor to vehicle speed and tractive effort. This gives the following data for the drivetrain, assuming the motor to be ideal:

Overall gear ratio, motor to wheel	12:1
Maximum motor displacement	9.5cc/rev
Rolling radius	199mm
Tractive effort with 200 bar of fluid pressure (full motor displacement)	1822N
Motor displacement per metre (full motor displacement)	91.2cc/m
Volume stored in hydraulic compliance at 200 bar (at 10.8 bar/cc)	18.5cc
Vehicle distance equivalent to volume stored in compliance	0.201m
Equivalent linear spring rate = 1822/0.201	9064 N/m
Mass of the vehicle including driver and instrumentation	450kg
Undamped natural frequency of vehicle mass and hydraulic compliance $=\sqrt{(9064/450)}$ rad/s	0.714Hz

Figure 6-8: Table of parameters describing linking of mechanical and hydraulic domains

6.4 Identification of important system parameters

6.4.1 Compliance and leakage

A fundamental parameter of the hydraulic system is the stiffness of the fluid line between the pump and the motor, a result of both the bulk modulus of the oil and the stiffness of the hose. The stiffness was found experimentally by blocking the fluid connection to the motor, and delivering fluid pulses into this closed volume. As the DDP injected fluid pulses into the line, the pressure was seen to rise until the system relief valve cracked, limiting the pressure. Due to the accurate calibration of the DDP, the volume injected as a result of each pulse was accurately known. The pressure rise caused by a known number of pulses, allowed the system stiffness to be obtained.

The pressure trace for this experiment shows three clear phases: pump-up, pressure relief and leak-down:

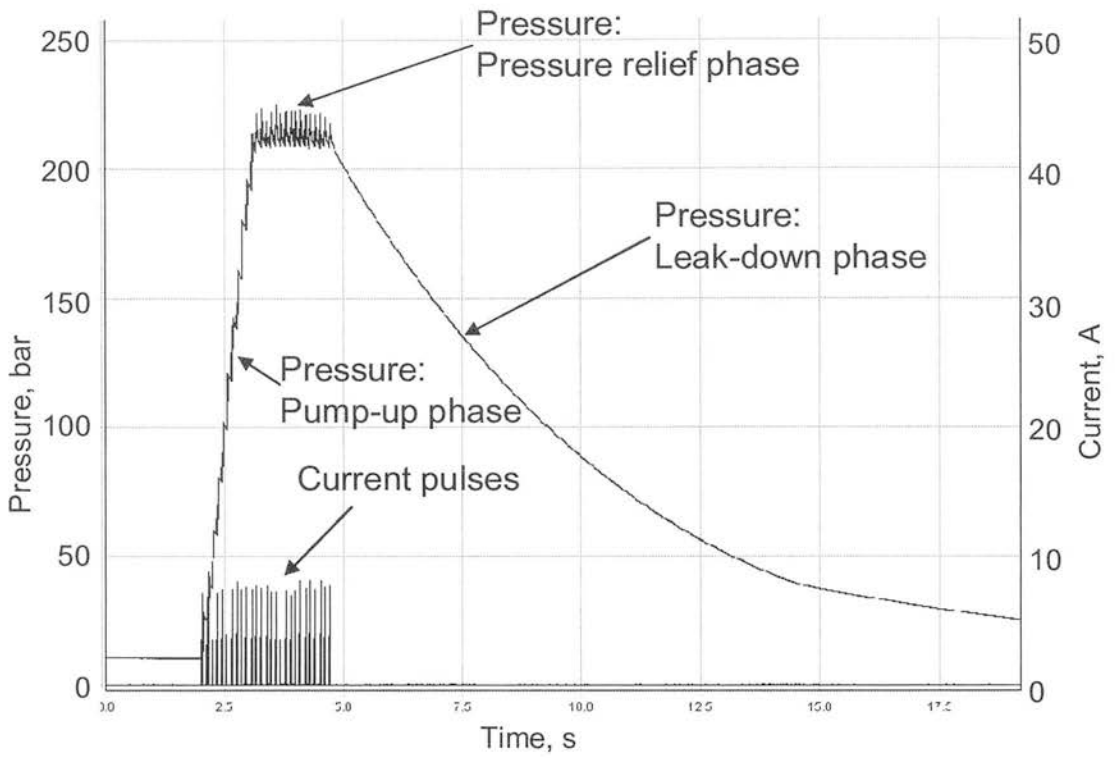


Figure 6-9: Test used to estimate hydraulic system compliance and leakage

During this test the motor was isolated from the system pressure. The stiffness calculation was complicated by the leakage that was observed from the closed fluid volume – thought to originate mainly in the relief and directional valves. From inspection of the leak-down trace, it takes 10 seconds to leak that which was delivered by the pump in 1 second, so it is estimated that in the 1 second it took to pump up the volume, approximately 10% of the fluid injected was lost due to leakage.

A detailed examination of the pump-up phase allowed the effect of the individual fluid pulses to be clearly seen:

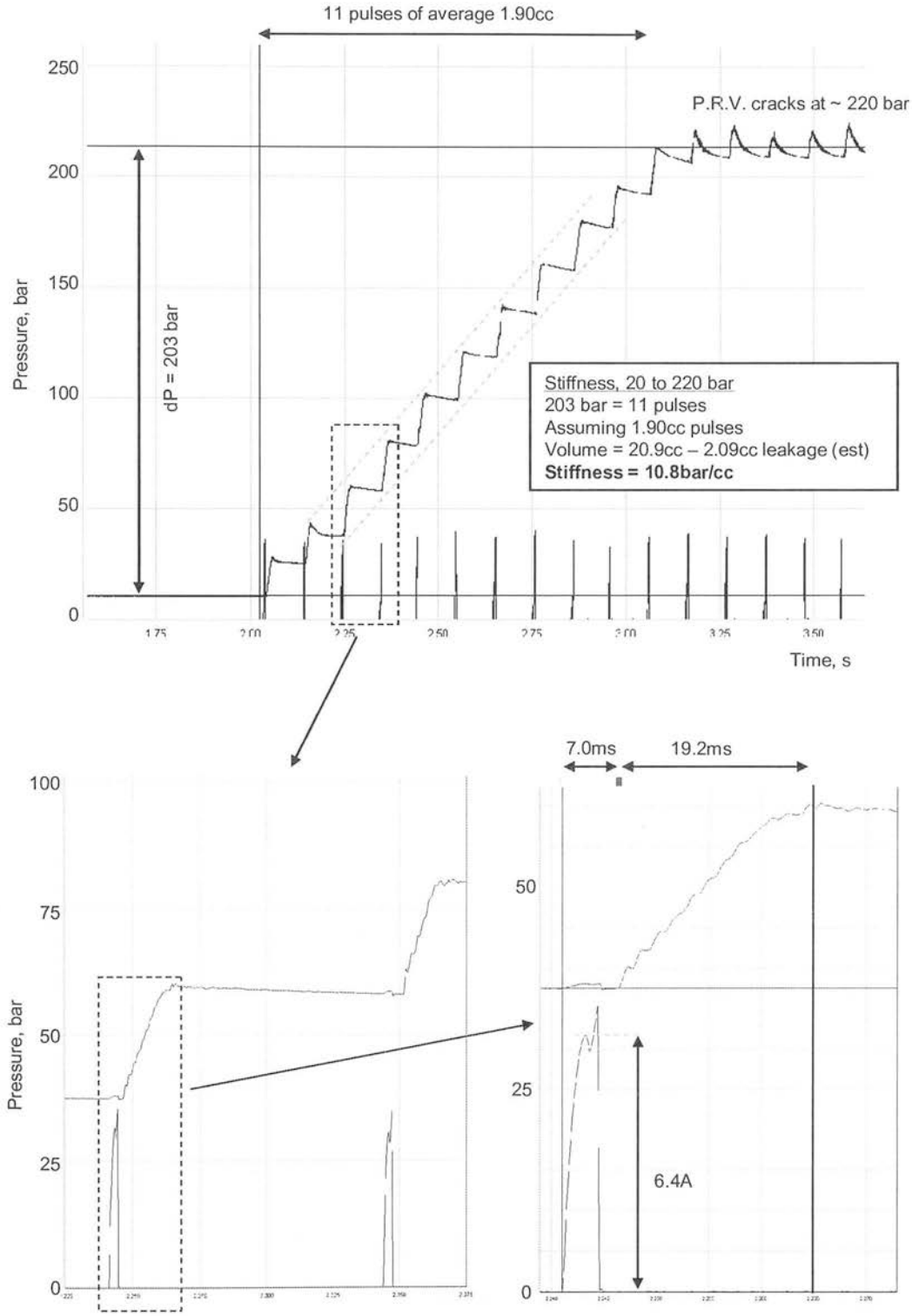


Figure 6-10: Experimental investigation of the compliance of the dead-headed vehicle hydraulic system, excited by full pumping strokes from the DDP at 1/3rd enabling fraction

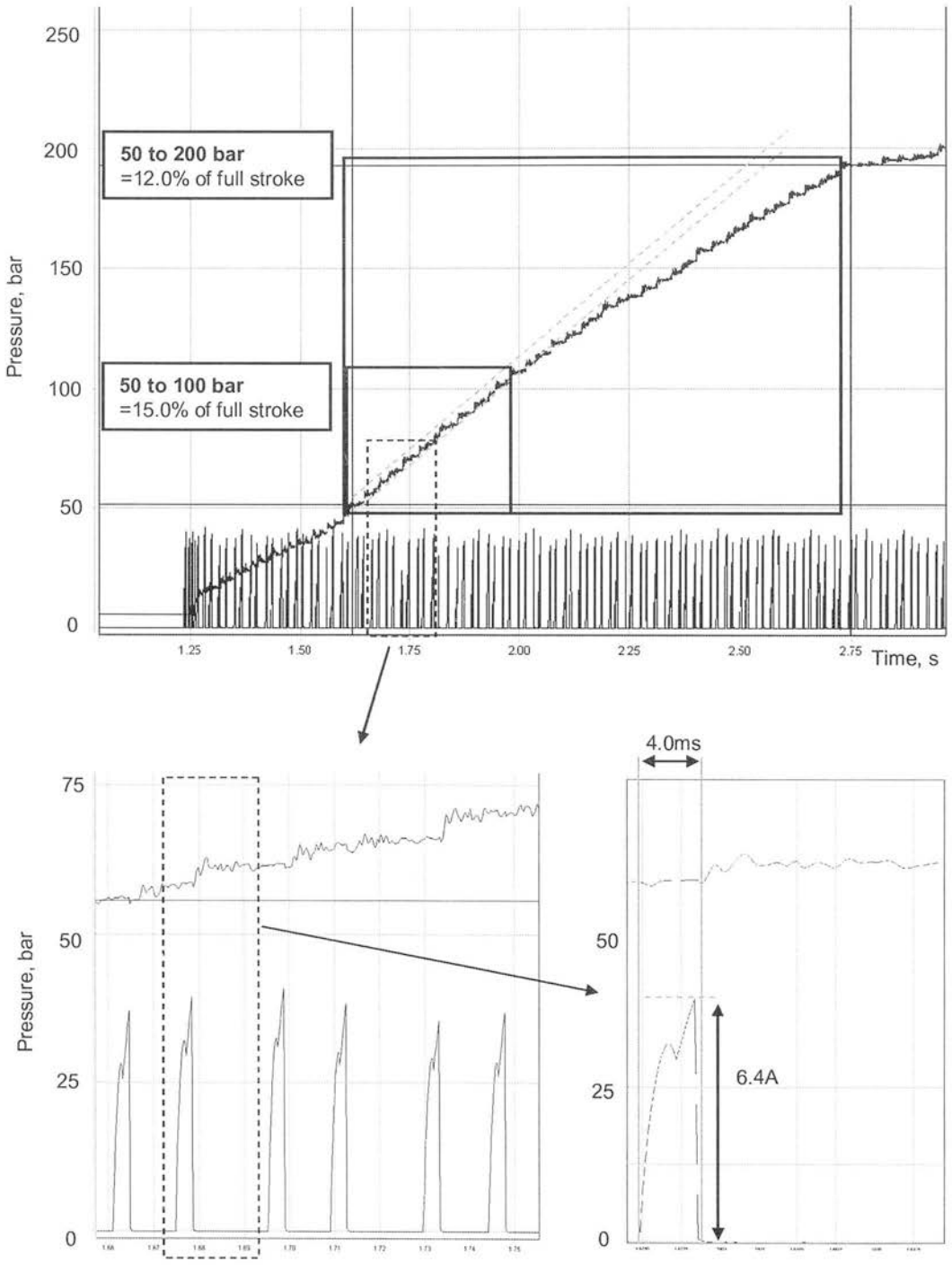


Figure 6-11: Results from the dead-headed system compliance test, excited with nominal 17% part strokes, at close to full enabling fraction

The effect on the system of each individual full pump pulse can be clearly seen to be correlated with the solenoid current trace. However the part strokes were small enough that it took over 130 of them to pump up the volume from zero to full pressure, and the effect of each individual pulse was difficult to discern. The effect of fluid compressibility on the part strokes is clear from the reduction in gradient as pressure rises.

6.4.2 Calibrating the Dymola model with above results

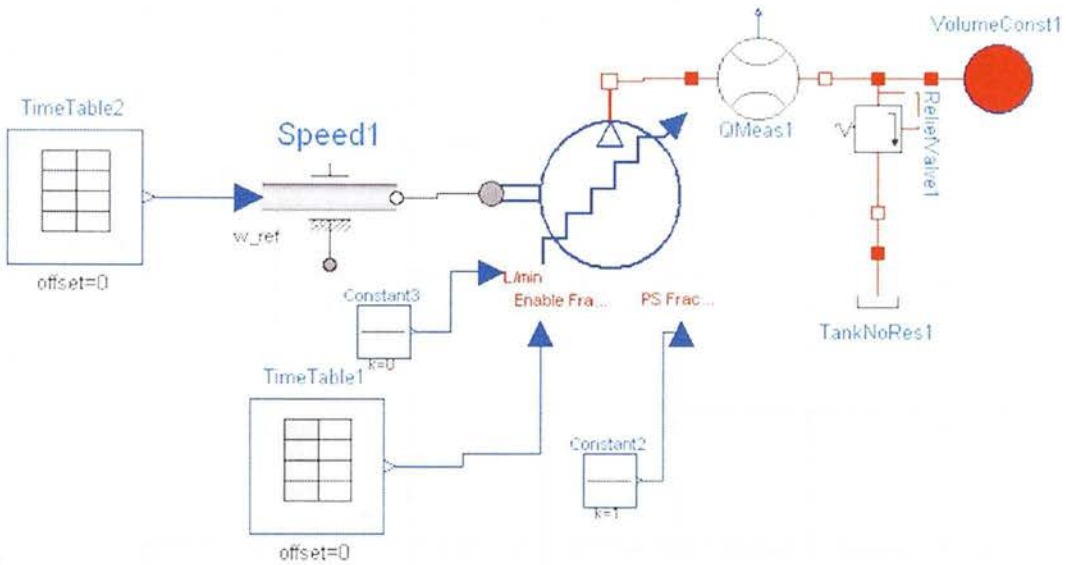


Figure 6-12: Simple Dymola model of the DDP pumping into a closed volume with a relief valve

A simple model of the external circuit during the above tests was created including a linear compliance (assuming constant bulk modulus), and a relief valve model containing a leakage proportional to pressure, and a further leakage which was proportional to the pressure above a cracking pressure.

From the number of full pulses needed to raise the system pressure to 203 bar , the estimate of leakage during this period of 10%, and the known volume displaced by each full pump of 1.90cc, the stiffness of the system was calculated to be 10.8 bar/cc, equivalent to an oil volume of 1.5 litres with a bulk modulus of 16000 bar. This was entered into the Dymola model and the parameter of leakage inside the relief valve model was varied to match the

observed pressure decay. A good match was achieved with a laminar leakage of 0.001 l/min per bar; this leakage was modelled as coming solely from the relief valve. The results from this simulation are shown below overlaid on the experimental results:

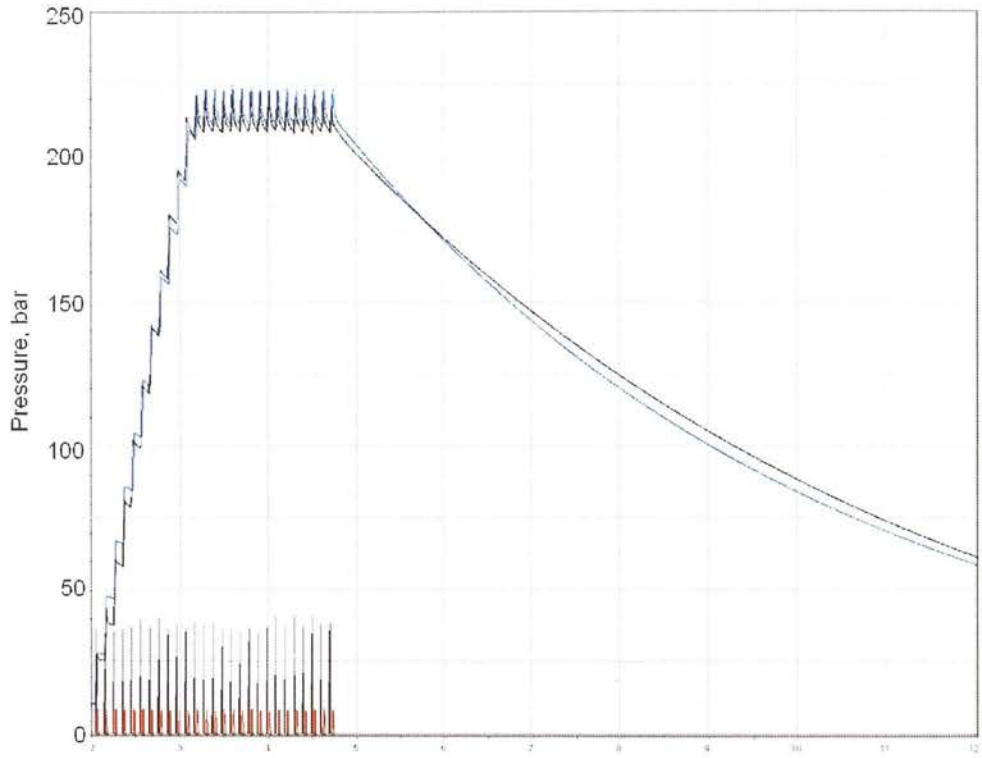


Figure 6-13: Identification of model parameters for system compliance and leakage - comparison of experimental pressure and current (black) and simulation (blue=pressure, red=DDP flow).

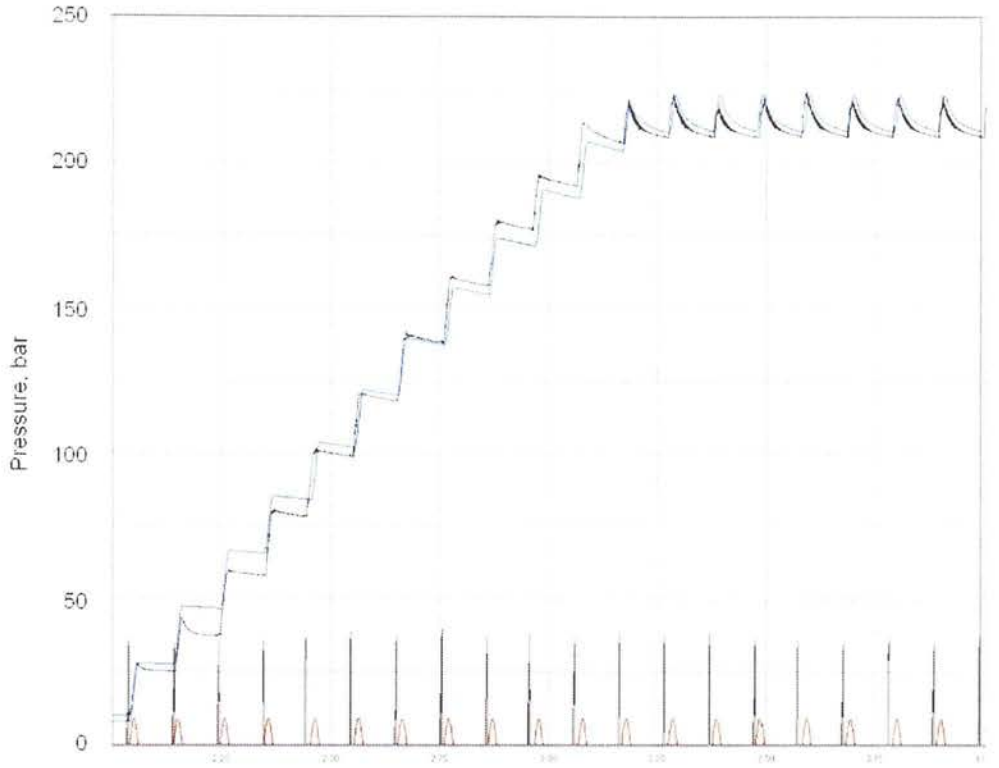


Figure 6-14: Detail of above figure in the "pump-up" phase

The simulation displayed similar behaviour to the experiment when excited with full strokes.

Once the motor and its flexible hoses were connected to the circuit, it was observed in experiments that there was significant extra compliance and leakage. These were estimated by repeating the above experiment with the motor in the fluid circuit, but locked from rotating by clamping the drive pulley. The total system stiffness was observed to reduce to 9.4 bar/cc (equivalent to a closed volume of 1.7 litres of oil with bulk modulus of 16000 bar), while leakage increased dramatically to 0.0055 l/min per bar.

It should be noted that the leakage behaviour of the Lucas motor is probably more complicated than a simple laminar leakage, particularly with respect to speed and temperature variations. However a full characterisation would have involved testing the motor on a dynamometer rig, which was not available.

The parameters which were identified during this process are listed below.

Volume displaced by a full pump	1.90cc
Equivalent compliant volume (assuming bulk modulus 16000 bar)	1.70 litre
Relief valve cracking pressure	220 bar
Relief valve leakage above cracking pressure	1 litre min^{-1} bar^{-1}
Relief valve leakage across full pressure range (includes leakage from all other valves external to the DDP)	0.0010 litre min^{-1} bar^{-1}
Additional leakage from motor	0.0045 litre min^{-1} bar^{-1}

6.5 The DDP model

The DDP model used in this work was a development of Ehsan's concept of a "function generator", which produced ideal half-sinusoids of flow. Ehsan's model was not suitable for use in this work for the following reasons:

- it could not create partial strokes;
- it did not account for fluid compressibility, thought to be the dominant non-ideal effect when part strokes were to be considered;
- it was implemented in the Simulink environment, which did not offer pre-defined hydraulic or mechanical building blocks

From the outset, it was desired that the new model have two modes, namely:

Ideal pump approximation. In this mode the discrete DDP pulses were not modelled and the output flow was treated as the product of shaft speed, total displacement and the present displacement fraction demanded, with a modification to account for compressibility effects. Because it required much less computation (approximately one tenth), this mode was useful for rapid development of the overall system model.

Detailed DDP model. In this mode, the DDP pulses were modelled in detail including the effect of compressibility on the shape of the pulses. This mode allowed the effect of the DDP pulses on the system to be investigated.

6.5.1 Ideal pump 'wrapper'

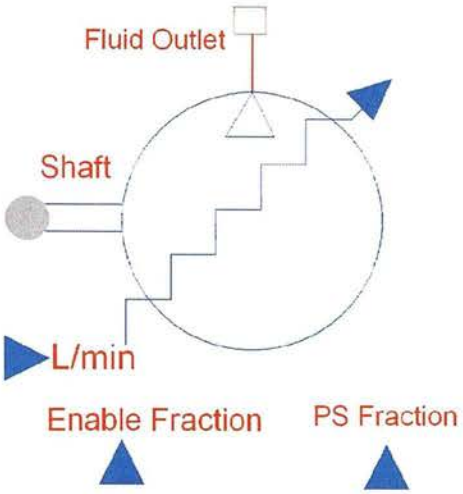


Figure 6-15: Icon for the DDP/Ideal pump model

The detailed DDP model was contained within a 'wrapper' model selectable between 'ideal' and 'DDP' modes. The pump is driven by the circular grey rotational mechanical connector – linking the speed and torque of the pump to that of the prime mover. The square red fluid outlet connector provides a flow output to the external hydraulic system, the resulting pressure in the system being fed back. There are three solid blue triangular control signal ports; unlike the fluid and mechanical connectors, these are unidirectional much like connectors in Simulink (in Dymola, solid blue ports are signal inputs, and outlined blue ports are signal outputs).

The 'L/min' port allows a controller to supply an absolute flow rate demand in l/min, the displacement fraction of the pump being calculated taking into account shaft speed and maximum displacement, such that the output flow rate corresponds to the demand (before the effect of compressibility is applied). The 'Enable Fraction' port allows an external controller to directly demand pump displacement fraction (0 to 1). The resulting displacement demand from both of these ports is summed inside the model, but in all subsequent work the 'L/min' port was set to zero and the pump was controlled with a displacement fraction demand.

The 'PS Fraction' port allows an external controller to demand partial strokes, the DDP delivering a fraction of a full pumping stroke (0 to 1). The phase angle of the closure of the LPV is calculated inside the model to achieve this fraction of full stroke. For simulating the DDP propel demonstrator, this port was fed with a constant of 0.166 (=1/6); this reflected the fact that the real DDP controller was capable of delivering only full strokes or 1/6th part strokes.

The model has a number of parameters:

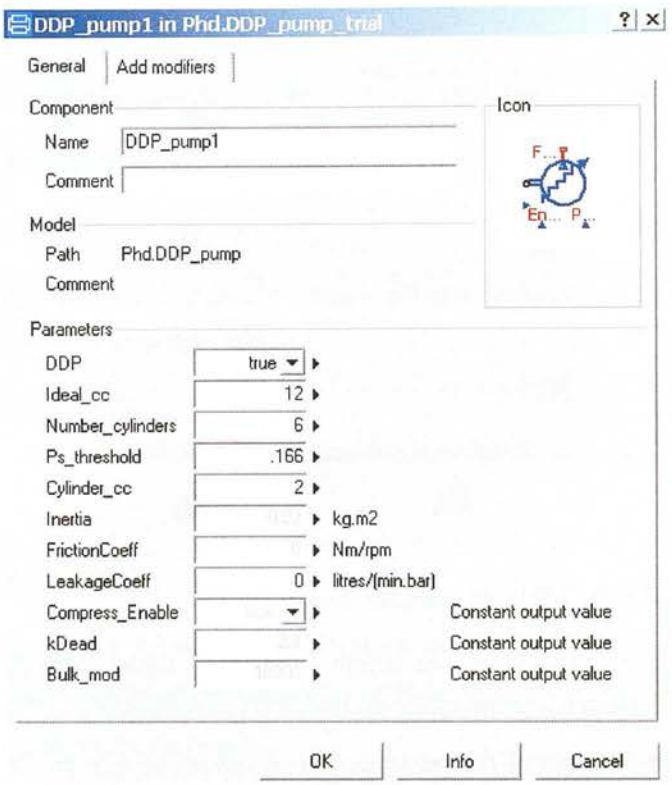


Figure 6-16: Parameters for the DDP/Ideal pump model

DDP selects whether the output of the detailed DDP model is used (=TRUE) or the ideal pump approximation (=FALSE).

Ideal_cc is the maximum displacement of the pump in ideal pump mode.

Number_cylinders is the number of cylinders in the detailed DDP model, each of which is of size *Cylinder_cc*. These are assumed to be arranged at equal angular spacing around the shaft, so if 6 is entered in this box the DDP is arranged as 6 cylinders at 60° spacing.

Ps_threshold is the fraction of full displacement demand, below which the controller in the detailed DDP model will deliver partial strokes of fraction determined by the input to the 'PS Fraction' port. The controller assumes that each partial stroke will deliver the fraction of a full stroke determined by *Ps_threshold*. Normally the 'PS Fraction' port will be set to the same as *Ps_threshold*, in which case the controller has an accurate estimate of the actual flow fraction delivered by a partial stroke. Separating the actual partial stroke delivered, from the partial stroke assumed by the controller, allows the investigation of the effect on the system performance of an inaccurate controller estimate of the partial stroke actually delivered- this could be caused by variation in the closing time of the LPV, for instance.

Inertia allows the pump inertia and other rigidly-coupled inertias to be modelled.

FrictionCoeff allows speed-related loss torque loss to be added. For the modelling of the propel demonstrator, the known idle losses of 120W at 1500rpm gave a value of 0.00051 Nm/rpm. In fact the real idle torque losses of the DDP are not simply first order with respect to speed, but this was an adequate approximation given the limited range of engine speed due to the presence of the engine governor.

LeakageCoeff allows a parasitic leakage (upstream of the HPVs) to be modelled; this was set to zero, because the HPVs can be assumed to be leak-free if the DDP is working properly.

Compress_Enable enables the calculation of the reduction of effective output flow due to internal pump compressibility. If *DDP*=TRUE, this is modelled in detail with the correct effect on partial strokes; otherwise the ideal pump model assumes that full strokes only are delivered. *Kdead* and *Bulk_mod* are the parameters for the compressibility modelling of cylinder dead volume ratio and fluid bulk modulus respectively.

The compensation for the effect of fluid compressibility is not built-in to the pump control, but is applied to the output flow rate as described later. Therefore the actual flow rate produced from the fluid outlet is reduced compared to the demand, depending on the pressure at the fluid outlet.

The interior detail of the DDP/Ideal pump model is shown below:

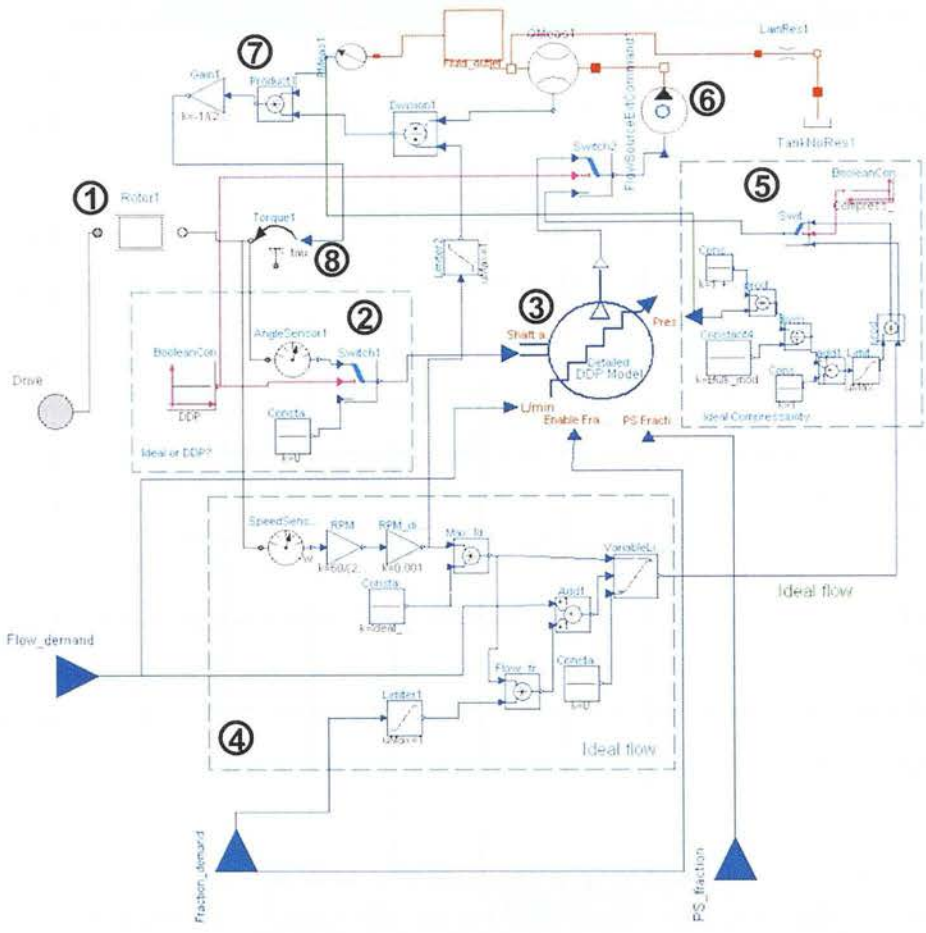


Figure 6-17: Detail of the DDP/Ideal pump model

The drive shaft (1) is coupled to an inertia; in the simulation of the propel DDP demonstrator, the engine inertia was lumped with the pump inertia in this component. The speed-related drag torque is also be applied here. If *DDP*=TRUE, the angular position of the shaft is routed to the detailed DDP model via a switch (2); otherwise, zero is routed which means that all of the computationally-intensive events inside the detailed DDP model are inhibited. The detailed DDP model (3) is also fed with the flowrate demand, enabling fraction demand and part stroke fraction demand from the model input ports; a virtual pressure transducer creates a pressure signal from the fluid port for calculation of the compressibility effect. For the ideal pump mode, output flow rate is calculated from the input demands and the shaft speed (4), and compensation for compressibility is applied assuming full strokes (5). The signal to the ideal flow source (6) comes from either the detailed DDP model or the ideal pump depending on the value of *DDP*. The torque exerted by the DDP is calculated from (flow/speed)*pressure and applied to a torque source (8); aside from the fixed speed-related losses, the mechanical efficiency is assumed to be 100%. Note that if *DDP*=TRUE, the shaft

torque pulsations reflect the instantaneous flow and pressure pulsations at the fluid outlet; otherwise the shaft torque is smooth.

6.5.2 Detailed DDP model

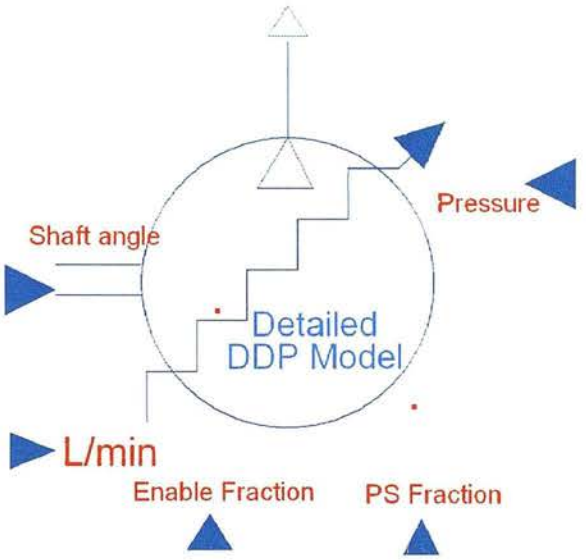


Figure 6-18: Icon of the detailed DDP model

The detailed DDP model has the same signal port definitions as for the 'wrapper' model described above. However the mechanical and fluid connections are single-direction signals rather than bi-directional connections.

The detail of this model is shown below.

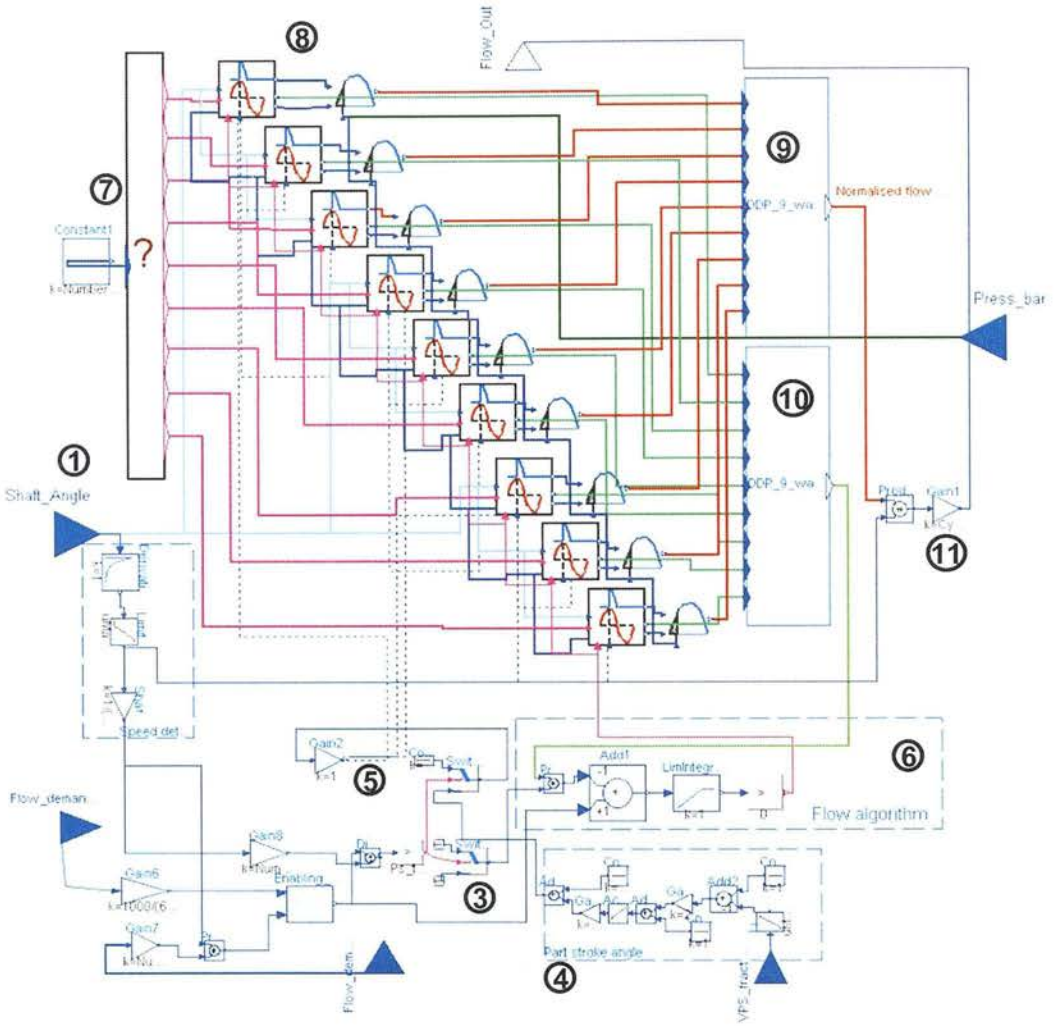


Figure 6-19: The detailed DDP model

The shaft angle (1) is routed to each of the single-cylinder models, each of which has its own offset phase angle calculated to space all enabled cylinders evenly around the shaft. The derivative of angle yields the speed of the shaft, which is also used to calculate the displacement fraction demand from the 'L/min' port. This is summed with the displacement demand fraction from the 'Enabling Fraction' port to give the final pump displacement demand. If this is less than *PS_threshold*, then the pump is operating partial stroke mode and the actuation angle calculated in (4) is routed (5) to the single cylinder models (8). The actuation delay time is assumed to be zero, i.e. the LPV closes the instant the command to do so is received. In the real DDP, the controller must compensate for the finite actuation delay by firing the solenoid FET in advance of the desired closing angle; for the purposes of the simulation it is assumed that the controller is perfectly calibrated with this delay, such that the LPV always closes at the angle desired.

The DDP is modelled as up to 9 separate cylinders, each spaced equally around the shaft. If *Number_cylinders*<9, then the unused single-cylinder models are disabled by the Boolean signals from (7).

The flow algorithm (6) sends a pulse request signal to the single-cylinder models, indicating that the accumulated difference between demanded displacement and delivered displacement is positive. If this signal is true when the closing event of the single-cylinder model occurs, then the LPV of that single cylinder is assumed to be closed and a decrement pulse is sent back to the flow algorithm via a summing block (10). This decrement pulse is weighted with the presently demanded part stroke fraction; therefore when the DDP is in part stroke mode, the pulse request signals are more frequent for a given displacement demand than when it is in full stroke mode.

The outputs of the single cylinder models are summed to give the flow rate output (9), normalised such that the magnitude of the half-sinusoid full stroke pulses is 1. The actual flow rate is calculated (11) by multiplying normalised flow rate by speed and *Cylinder_cc*, and routed to the output port.

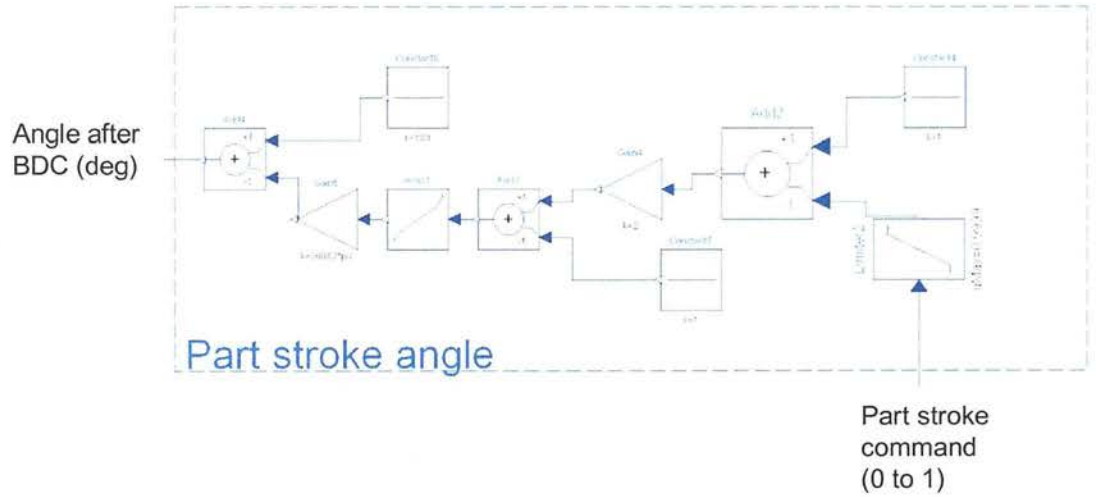


Figure 6-20: Calculation of the LPV closure angle from the desired part stroke fraction

The LPV closing angle is calculated from the inverse cosine of the desired part stroke fraction (0 to 1) as shown above.

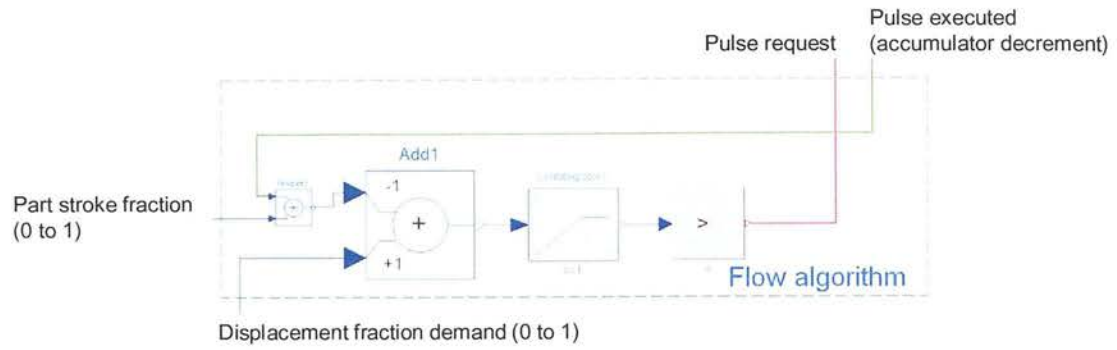


Figure 6-21: The DDP flow algorithm

The flow algorithm is modelled as a simple limited integrator, which accumulates the error between the desired displacement fraction and the actual delivered flow pulses. If this error becomes positive then the pulse demand signal goes TRUE, and a flow pulse will be delivered when the next available single-cylinder model reaches its decision point.

In fact the flow algorithm in the real DDP controller does not accumulate error continuously, but rather at each LPV firing event; this is 120 Hz for a 6 cylinder DDP at 1500rpm. This is only significant if frequencies in excess of 60Hz are present at significant amplitude in the displacement demand signal. In this case, the modelled DDP will accurately accumulate the error, but the real DDP controller suffers from aliasing. This is not significant for the propel demonstrator studied.

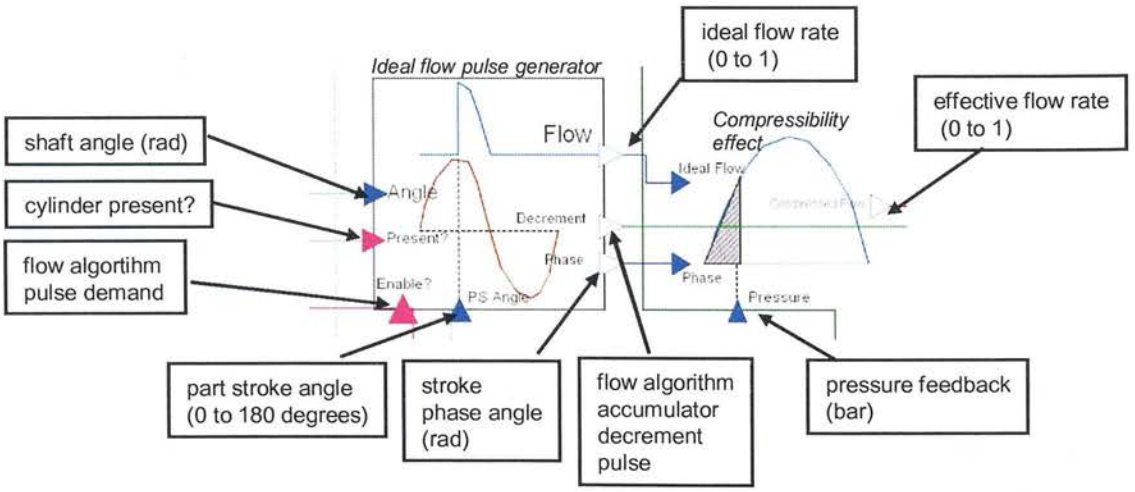


Figure 6-22: Generation of ideal flow pulses and the compressibility modification for each cylinder

The single cylinder DDP model is show above. The ideal "flow pulse generator" has an internal sine wave generator, corresponding to the normalised piston flow rate. Unless given a pulse demand, the flow output from this block is zero, simulating the DDP "idle" condition. When the shaft angle reaches the *PS Angle*, the value of the pulse demand signal from the

flow algorithm is examined; if TRUE, then the LPV is closed, a decrement pulse is sent to the flow algorithm, and the sine wave generator is connected to the output. When the normalised piston flow sine wave reaches zero (i.e. the TDC position) the LPV is de-latched, the flow output goes to zero and the block is reset waiting for the next revolution.

The flow pulse from the ideal flow pulse generator is routed to the compressibility effect block. At the moment that the flow pulse begins, the normalised volume of the stroke lost to fluid compressibility is calculated. The normalised volume to be compressed before useful flow is delivered (termed the "lost stroke"), is calculated from the remaining swept volume plus the dead volume:

$$Lost\_stroke = \frac{P}{\beta} \cdot (k_{dead} + \frac{\cos(phase)+1}{2}) \tag{54}$$

Where *phase* is the angle after BDC at which the LPV was closed.

The flow from the ideal flow pulse generator is integrated until this lost stroke has been displaced by the piston; after this point any further flow is passed to the fluid outlet. The re-expansion of the compressibility in the cylinder after TDC is not modelled, because it is of no relevance to the system connected to the fluid port. It is assumed that the compressibility energy is returned to the shaft after TDC with 100% efficiency.

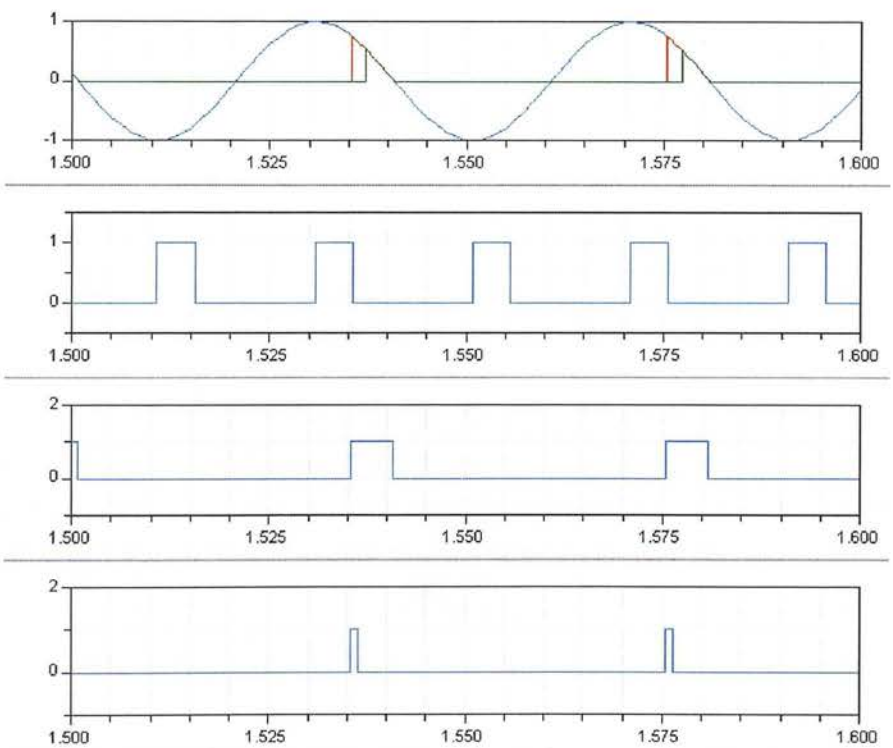


Figure 6-23: Single cylinder flow simulation; traces numbered from the top. Trace 1: blue=normalised piston flow rate, red=pre-compressed flow rate, green=compressed

flow rate; Trace 2: Pulse demand (1=pulse demanded); Trace 3:LPV state (1=closed); Trace 4: decrement pulse to flow algorithm

The operation of the single cylinder model is shown above with the compressibility effect active. It can be seen that at t=1.535s, the LPV was fired and a decrement pulse created. The red trace shows the partial stroke that was displaced, and the green trace the actual useful flow which was displaced into the fluid outlet.

Effect of compressibility of outlet flow

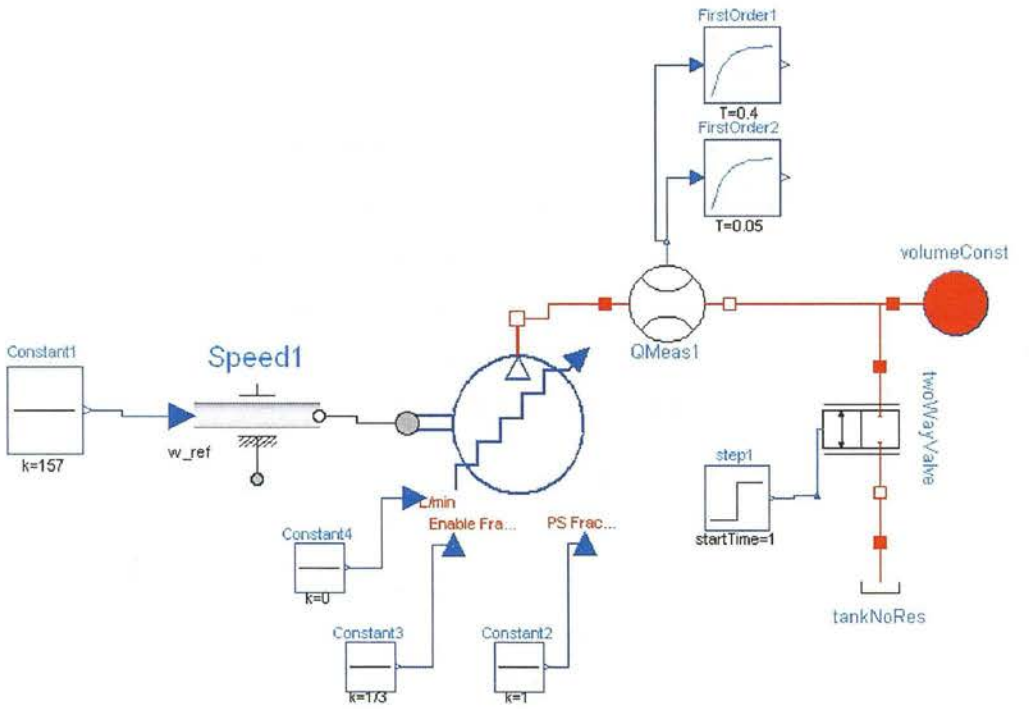


Figure 6-24: DDP/Ideal pump test model

The behaviour of the DDP model was investigated in a test model shown above. The DDP in the golf buggy is simulated (6 cylinders of 2cc each), running at 1500rpm. The compressibility compensation is enabled (k_{dead} =2.8, β=16000 bar). The DDP is commanded with a 1/3rd displacement fraction demand, this results in two opposite cylinders permanently enabled. The flow output enters a closed compressible fluid volume (v=0.2 litres, β=16000 bar) via a perfect flow transducer, the signal of which is filtered with a first-order low-pass. The two-port two-position valve begins in the open position, venting pressure to tank. At t=1s, this valve closes instantaneously and the flow from the DDP is trapped in the fluid volume, causing pressure to be raised. As the pressure rises, the compressibility effect in the DDP causes the flow to be reduced, leading to an asymptotic pressure trace.

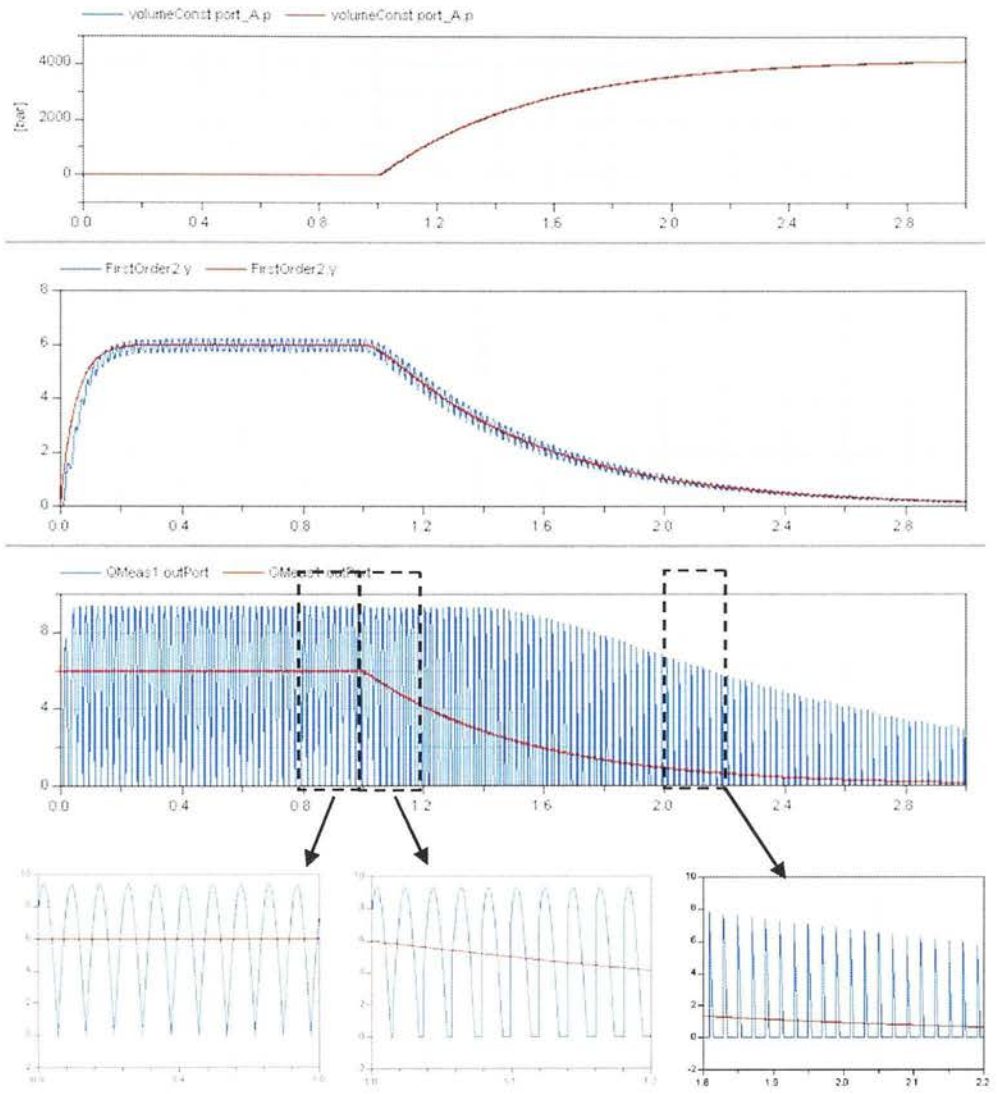


Figure 6-25: Results from the DDP/Ideal pump test circuit with pump compressibility enabled; red=ideal pump, blue=Detailed DDP

First, the ideal pump mode was compared to the detailed DDP mode, with compressibility enabled in both cases. The DDP was operating in full-stroke mode. The figure above shows that the pressure asymptotes in both cases at around 4100 bar; at this pressure the compressibility effect is so large that zero effective flow is created, although obviously in reality, something would break before this pressure was achieved. The flow pulses from the DDP start as ideal half-sinusoids, becoming more distorted as pressure rises and more of the piston stroke is absorbed in compressing the volume in each DDP cylinder.

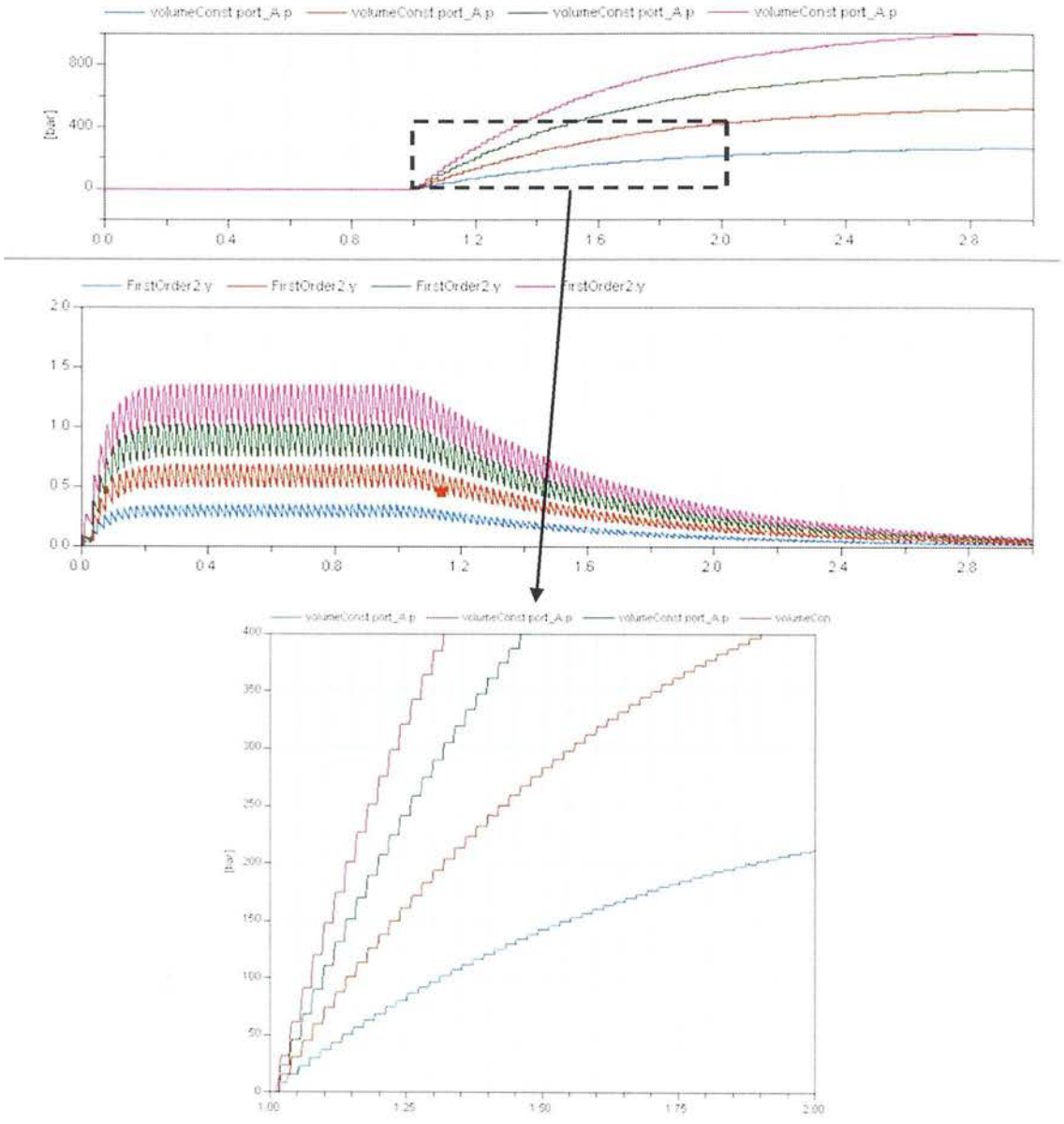


Figure 6-26: Results from the DDP model with compressibility (1500rpm; 6 cylinders; 1/3rd enabling fraction) pumping into a closed dead volume with different part stroke fractions (pink=0.20, green=0.15, red=0.10, blue=0.05); Trace 1: pressure; Trace 2: flow rate filtered with first-order low pass (τ=0.05s); Trace 3: pressure zoomed-in

The figure above shows the effect of fluid compressibility on small part strokes and underlines the importance of simulating the fluid compressibility effect at small part strokes. With very small part stroke fractions (0.05), the actual flow delivered is very strongly affected by pressure over the operating range of a propel transmission, as shown by the asymptotic shape of the blue curve in the lower trace of pressure.

Effect of compressibility on flow algorithm linearity

The DDP used in the propel demonstrator was calibrated to produce 17% part strokes. To investigate the effect of system pressure on the linearity of the flow output with respect to demand, the simulation model above was used with the closed fluid volume at the DDP outlet replaced with a constant pressure. The DDP model was set up with *Ps_threshold* = 0.17, while the "Ps Fraction" command was set to 0.18. The DDP was given a displacement demand ramp from 0% at t=1s to 100% at t=10s. The flow pulses from the DDP were filtered with a first-order low-pass (τ=0.05) so that the effect of individual pulses did not drown out the underlying average.

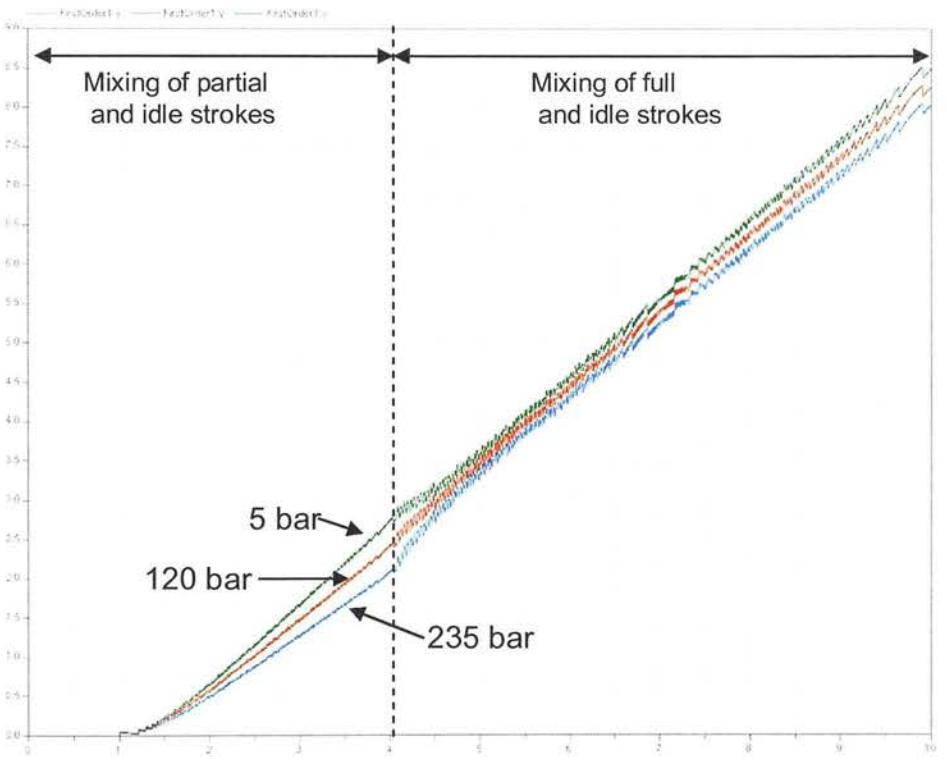


Figure 6-27: Effect of compressibility on flow linearity

The results above show that at 120 bar, the average DDP output flow is fairly linear with respect to demand. However at 235 bar, there is a distinct upwards kink at the transition to full strokes, while at 5 bar there is a similar downwards kink. Whether these "kinks" are significant or not depends on the properties of the hydraulic system. In a system with an overcentre valve, a sudden reduction in flow could cause the motor to decelerate suddenly even though the flow demand was ramping up.

Notes on compressibility

It should be noted that the k_{dead} of this machine of 2.80 is very large compared to typical values for axial-piston pumps of 0.1 (Dorey 1988). In axial piston pumps, this value is minimised by the designer as a prime objective because all compressibility energy is lost. This is not true for a DDP, as shown by the efficiency results of the C2 machine (Section 2.8.1), so the negative effect on overall efficiency of dead volume in the DDP, is very much reduced compared with the axial piston pump.

It may still be considered desirable to reduce the dead volume to achieve better calibration of output flow with respect to pressure at small part stroke fractions. However this dead volume also serves as the dynamic de-coupling compliance between motion of the piston and flow in the valves. It may be that reducing the dead volume of a DDP has the bad effect of increasing pressure transients inside the cylinder during commutation, causing an increase in structural and fluid-borne noise – this has been observed in high-pressure check-ball pumps for diesel fuel with very low dead volume (Nguyen-Schaefer 1998)

A large dead volume causes the output flow of the DDP to decrease substantially as the pressure rises. This characteristic has be termed by some authors as the 'stiffness' of the fluid supply; the flow from a perfectly stiff supply does not reduce as pressure increases. In systems which include an overcentre valve for instance, it has been shown by linear analysis that instability is worst with a perfectly stiff flow source, and improves as the flow source becomes less stiff (Andersen et al. 2005). Therefore a large dead volume in the DDP cylinder may help to stabilise some systems.

With k_{dead}=2.80, the extreme pressure-sensitivity of the net flow produced by very small part stroke fractions indicates that the "variable part stroke" method of DDP flow control may lead to poorer absolute control of very low vehicle speeds. However the presence in the vast majority of vehicles of a "man-in-the-loop" means that perfect linearity between pump flow demand and vehicle speed may not be necessary. Indeed axial-piston propel pumps exhibit a similar pressure-dependence at small swash angles due to the dominant effect of parasitic pump leakage at small swash angles. In fact the reduction in vehicle speed experienced when an obstacle is encountered, with a constant small flow demand, provides valuable information to the driver that the load at the wheels has increased, and this behaviour may well be desirable to the driver. It is possible that a simple feedback of pressure to the DDP displacement demand could be implemented in software to enhance or reduce this behaviour.

6.6 Results with an ideal pump flow source

6.6.1 Test 1-1: Acceleration mode

Initial investigation of the hydraulic and mechanical system were made with the DDP model in the "ideal pump" mode, whereby the detailed DDP pulses are ignored. Experimental data was compared from Test 1-1, a fairly gentle sequence from rest to maximum speed and back again in "acceleration" mode. Results from the simulation are compared below with experimental data.

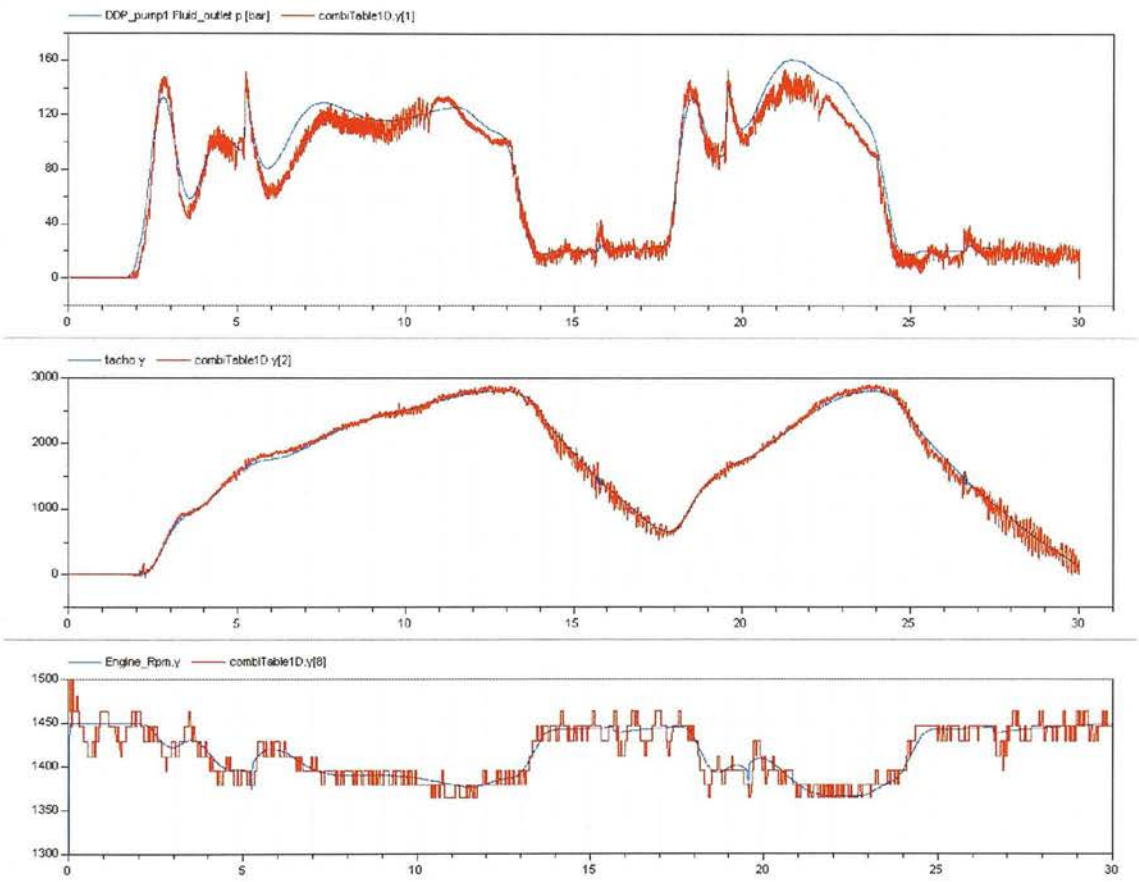


Figure 6-28: Simulation results with ideal pump model; blue=simulation, red=experiment; from the top: pump pressure, motor speed, engine speed

Obviously the high-frequency pulsation in the measured data is missing in the simulation, because the simulated DDP was a smooth ideal flow source, but the majority of the low-frequency system behaviour is reproduced well. The initial pressure spike at t=2.5 is due to accelerating the vehicle mass. The fundamental frequency of the low-frequency pressure oscillations during acceleration match fairly well; this indicates that vehicle mass and hydraulic compliance are fairly close to the actual values. The further spike at t=5.2 is due to the sudden change of displacement of the motor and synchronised reduction in pump

displacement; the size of this spike is very sensitive to the switching time of the motor displacement. The good match of steady-state pressure during the braking phase between t=13 and t=18 indicates that the setting of the overcentre valve model is accurate. However it is obvious that the experimental trace of motor speed has high-frequency oscillations during the deceleration phases, whereas the simulation is smooth.

The engine speed trace shows reasonable overall agreement, although the accuracy of the measured data is obviously limited by the quantisation inherent in measuring the period with 0.5ms accuracy once per rev.

This experimental run had rather gentle accelerations and therefore did not drive the system close to its performance limits.

6.6.2 Test 2-4: Speed mode

To investigate the effect of more aggressive driving, test 2-4 was simulated, a sequence of sudden step increases in speed demand and hence pump flow.

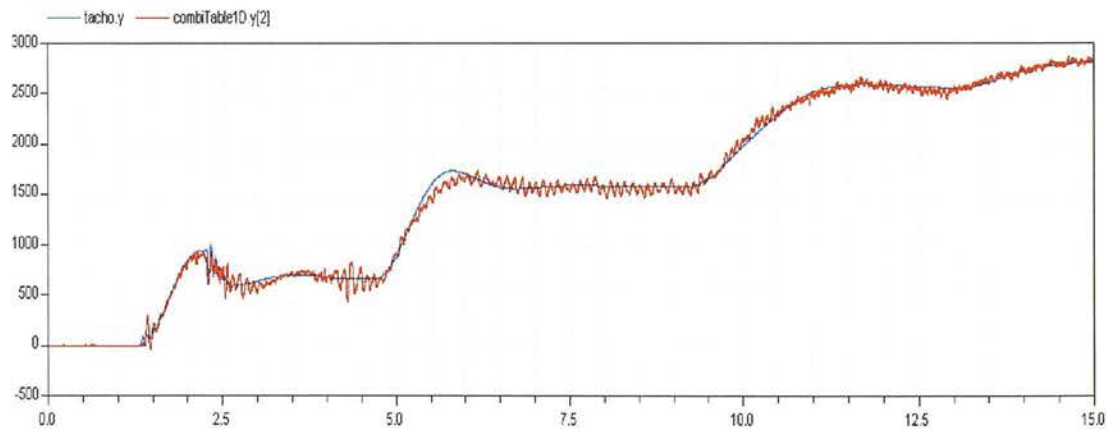


Figure 6-29: Test 2-4; motor speed; blue=simulation, red=experiment

The response of the motor speed to the step demands show that the effective damping factor of the fundamental second order system dynamics increases with speed, just as was observed in experiment. This is due to the higher leakage, motor friction and circuit pressure drop as speed increases.

The undamped first natural frequency of the system is determined by the vehicle mass and the hydraulic compliance. In applying a DDP to a vehicle propel system, some compromise must be reached between the compliance of the system and the acceptable pressure ripple inherent in the DDP principle. To investigate the effect of the natural frequency on the response of the vehicle to flow demand steps, test 2-5 was simulated with a range of compliances.

In order to draw conclusions which might be transferred to future DDP system implementations, it is useful to be able to characterise the system compliance in a scale-independent way. To help with this, the "Compliance Number" (CN) is introduced, being defined as the number of full cylinder strokes required to raise the system pressure from zero to the maximum pressure, with the motor locked and with zero system leakage. In the case of the experimental vehicle, this evaluates to 10.8.

Using the properties of the drivetrain given in Section 6.3.4, and the equation for undamped natural frequency of a second-order mechanical system of $\omega=\sqrt{(k/m)}$, these compliances give the following range of undamped natural frequency:

Fluid volume, litres (β=16000bar)	Compliance Number (full strokes to max pressure)	Expected natural frequency, Hz
0.4	2.6	1.37
0.8	5.2	0.97
1.6	10.5	0.68
3.2	21	0.48
6.4	42	0.34

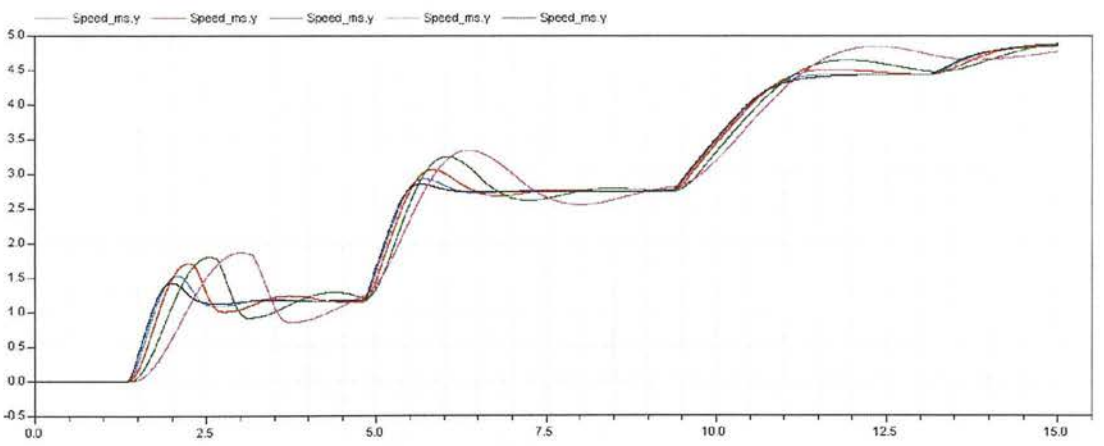


Figure 6-30: Test 2-4, response of vehicle speed to flow demand steps, with ideal pump and a range of system compliances. Black=0.4(CN=2.6), blue=0.8(CN=5.2), red=1.6(CN=10.5), green=3.2(CN=21), pink=6.4(CN=42)

As the system compliance increases, the natural frequency and the damping factor of the second-order system becomes lower, therefore the response to a flow demand step becomes more oscillatory. It should be noted however that a filter or s-curve between the operator input and the DDP flow demand will help to avoid oscillation. This can be seen by repeating the

simulation using the demand from Test 2-3, where the demand steps were passed though a low-pass filter (τ=2s) before being passed to the pump:

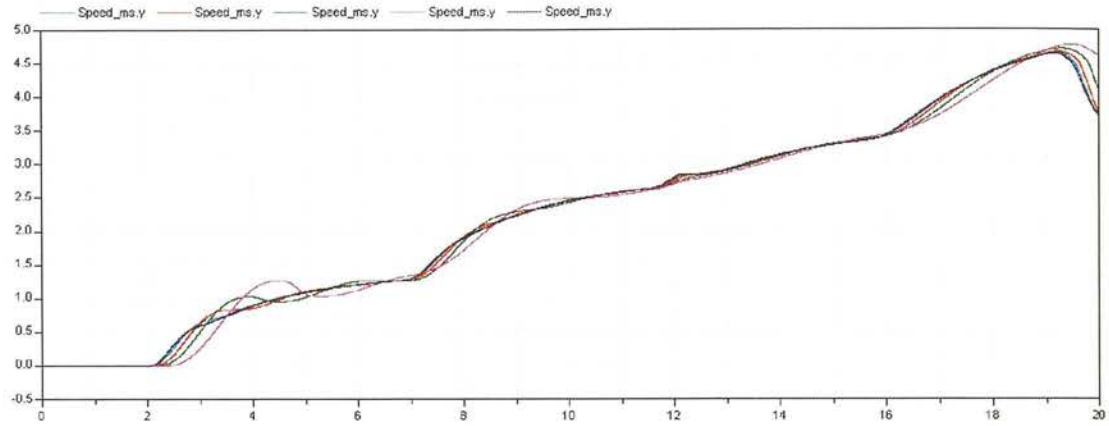


Figure 6-31: Repeat of above with Test 2-3, with displacement demand filter

Clearly the filter helps to prevent excitation of the fundamental frequency. In vehicles with modest requirements for dynamic propel performance (aerial work platforms, for instance), the combination of a compliant system and a low-pass filter or s-curve generator may provide acceptable control. However this approach is unsuitable for applications where stiff flow control is a key requirement, such as dual-path machines (e.g. a skid steer loader); here compliance must be minimised to prevent unacceptable yaw oscillations.

In the system studied here, control of vehicle speed is achieved by open-loop flow control. However if an automotive-style control is desired then the pump could be controlled in a pressure-control mode, which could also help control oscillatory behaviour. Some combination of flow and pressure control may be the optimum, and vehicle speed feedback may also be used. This deserves further work in the future.

As well as speed oscillations, high system compliance has a further undesirable effect. If the vehicle encounters a kerb, the pressure would build up until sufficient tractive effort is developed to climb the obstacle. Even if the driver suddenly reduced flow demand to zero as the kerb is mounted, the vehicle would continue to move forward for some distance due to the volume stored in the compliance; no conceivable control system of the DDP is able to overcome this undesirable effect, because a DDP cannot actively remove the fluid stored in the compliance.

6.6.3 Braking with the overcentre valve

Test 7-4 consists of a series of sudden position demand steps with the controller with the "Fast" range and the "Loose" s-curve filter active. This reveals the response of the system to a smooth acceleration and deceleration sequence, allowing the operation of the overcentre valve

to be investigated. For these tests the position sensor was fitted, allowing the overall motion control of the vehicle to be examined.

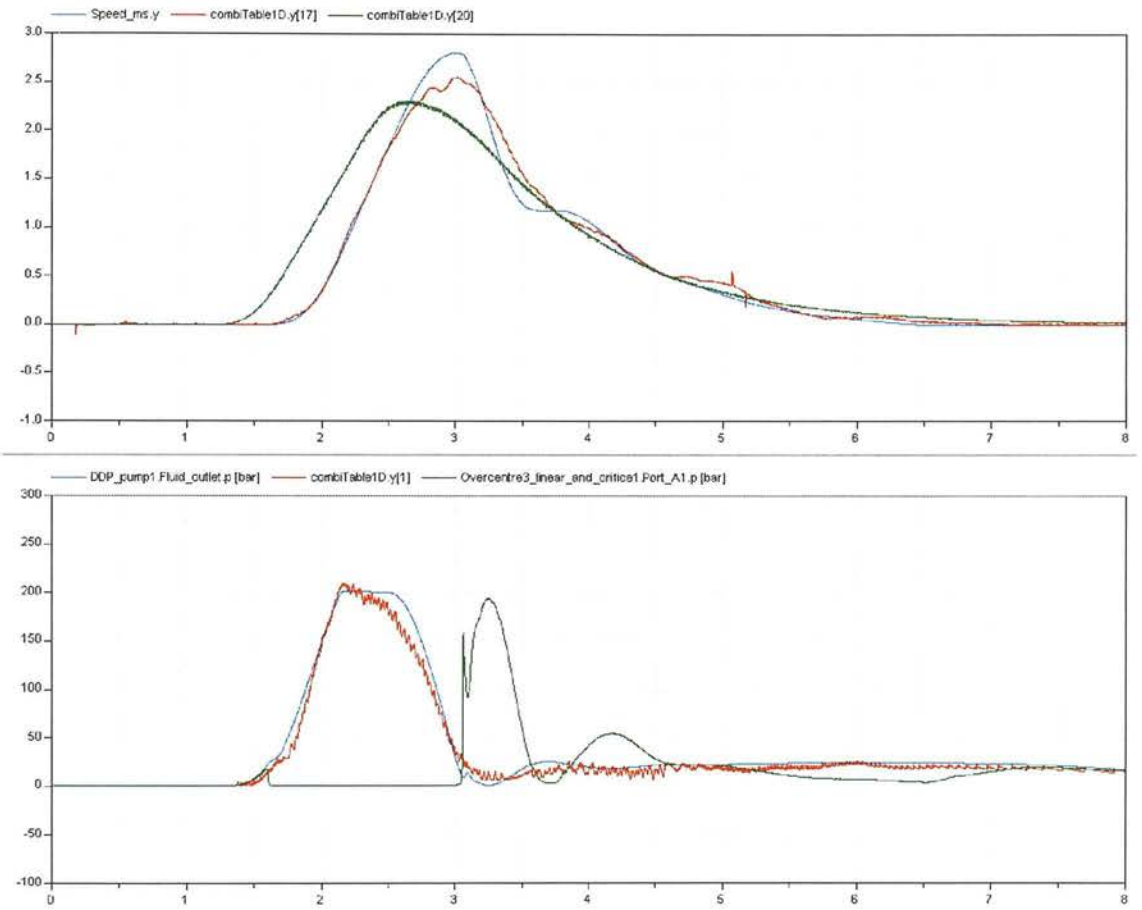


Figure 6-32: Test 7-4: position mode fast & loose, ideal pump; Top trace: vehicle speed: blue=simulation, red=experiment, green=ideal speed command; Bottom trace: blue=pump pressure (simulation), red=pump pressure (experiment), green=motor back-pressure from overcentre valve (simulation)

The top trace above shows the displacement demand signal sent to the DDP in the experiment, scaled to give the expected motor speed as if the hydraulic system were perfectly stiff with no leakage (green); this can be considered to be the speed demand of the vehicle.

During the deceleration from t=3s, the vehicle speed in the experiment (red) showed a pronounced low-frequency oscillation. Similar frequency and amplitude of oscillation is seen in the simulation (blue), although the phase of these oscillations is not aligned with experiment.

The lower trace shows that the pump pressure from experiment (red) settled to around 25 bar, expected from the pilot ratio (8:1) and cracking pressure (200 bar) of the overcentre valve.

Oscillations can be seen, leading the speed oscillation by about 90 degrees. The simulation showed similar results (blue). The simulated back-pressure created by the overcentre valve (green) is seen to be about 180 degrees out of phase with the simulated pump pressure oscillation; this is the classic behaviour of a hydraulic transmission with a mainly inertial load, braked by an overcentre valve (Overdiek 1981).

It is known that the main damping for these oscillations comes from leakage and pressure drops in the hydraulic system, and losses in the coupled mechanical system (Andersen et al. 2005). This oscillation behaviour is therefore very sensitive to the correct identification of these parameters, so the simulation results in this mode of operation cannot be expected to be perfectly accurate.

6.7 Results with a detailed DDP model

For comparison with the ideal linear pump model, Test 1-1 was simulated with the detailed DDP model. For the following results, the DDP was modelled with 6 cylinders each of 2cc displacement, k_{dead}=2.80, and zero internal leakage.

6.7.1 Comparison of DDP simulation and experiment: acceleration mode

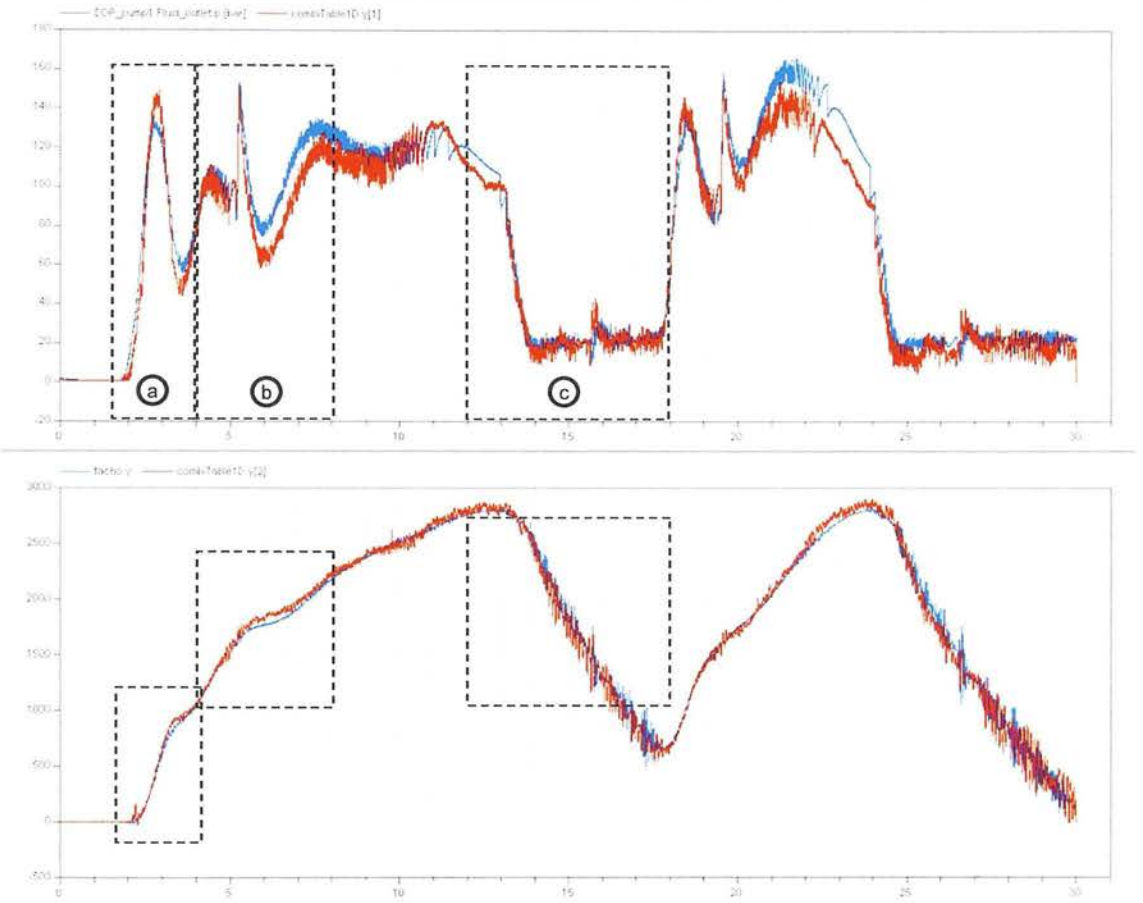


Figure 6-33: Test 1-1; Simulation results with detailed DDP model (blue=simulation, red=experiment); top trace=pump pressure; bottom trace=motor speed

The above figure shows a comparison between detailed DDP simulation and experiment. Overall the simulation shows good correlation with the experiment. Similar deviations were observed with the ideal pump simulation, indicating that they can be explained by imperfect identification of the hydraulic and mechanical system parameters.

More details of the regions marked above as "a", "b" and "c" are shown below:

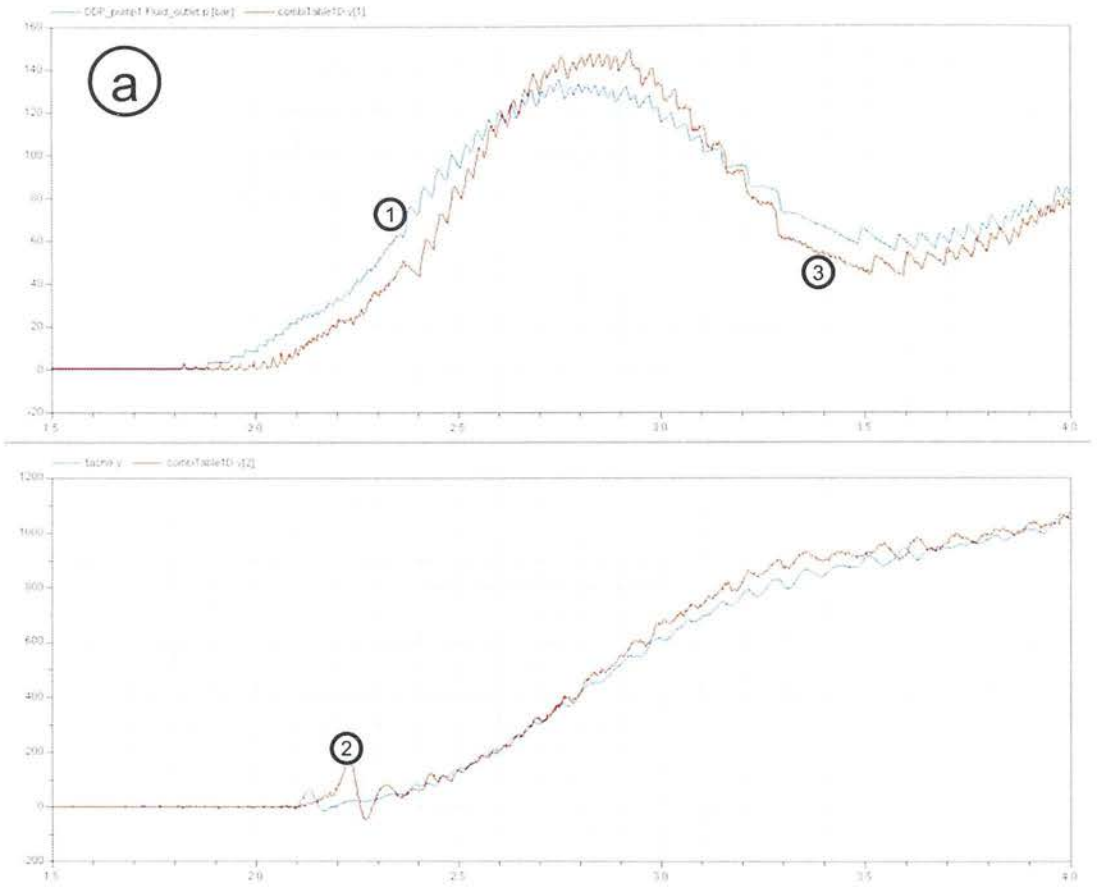


Figure 6-34: Detail of area "a" (blue=simulation, red=experiment); top trace=pump pressure; bottom trace=motor speed

Before the vehicle starts to move, the experiment showed a transient acceleration of the motor speed (2); a similar event is present in the simulation results, although the magnitude is much less. It is possible that in addition to the mechanical compliance identified between motor and vehicle mass due to the tyre, there was some backlash in the differential gear, which is not modelled in the simulation.

There is a clear increase in the magnitude of pressure pulsation when the DDP transitions from part strokes to full strokes (1), which is reproduced well in the simulation.

At point (3) the DDP operating point is 50% enabling fraction, mixing full and idle strokes (enabling sequence ="...0P0P0P0P0P..."). The DDP flow becomes momentarily smoother because exactly three pistons every revolution are enabled. As the displacement demand increases above 50%, extra pulses are introduced into the enabling sequence, causing increased pulsation. This effect is clearly reproduced by the simulation.

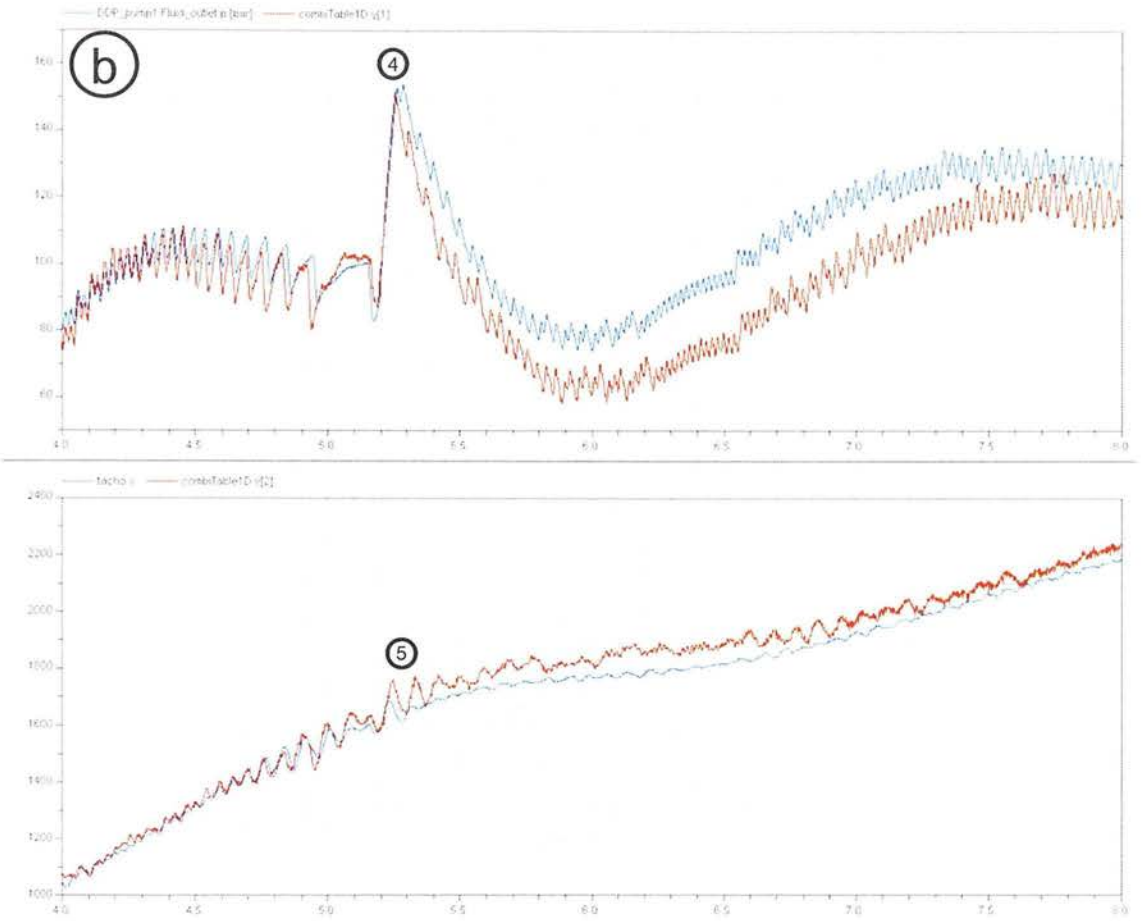


Figure 6-35: Detail of area "b" (blue=simulation, red=experiment); top trace=pump pressure; bottom trace=motor speed

As the vehicle accelerates, the displacement demand of the DDP increases until 100% enabling fraction is achieved, this can be seen from the decreasing frequency of negative pressure spikes caused by less-frequently disabled cylinders. The controller initiates the change to reduced motor displacement (4) causing a pressure spike, due to imperfect calibration of the controller for estimated the time taken for the effective motor displacement to reduce. There is a small spike in the motor speed at this point (5) but overall the vehicle acceleration is fairly continuous, and this transition could not be felt by the driver.

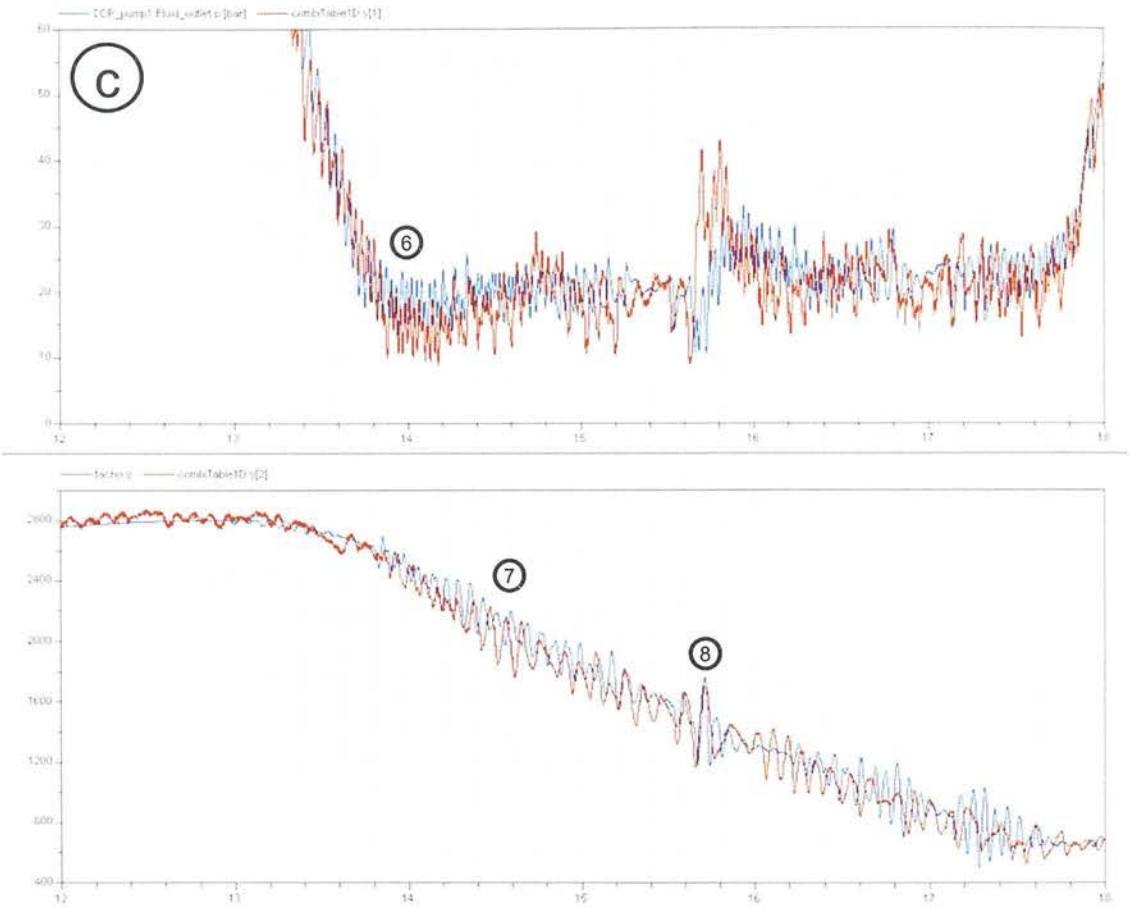


Figure 6-36: Detail of area "c" (blue=simulation, red=experiment); top trace=pump pressure; bottom trace=motor speed

As the displacement demand reduces, the pump pressure falls (6) until the back-pressure created by the overcentre valve starts to decelerate the vehicle. The oscillations on the motor speed (7) which were absent in the ideal pump simulation are clearly shown in the detailed DDP simulation. As the vehicle speed drops further the motor is switched to full displacement and the DDP displacement demand increases; this causes a distinct spike in the motor speed trace (8) but again no discontinuity could be felt by the driver at this transition.

6.7.2 Comparison of ideal pump and DDP simulation

The simulation results from Test 1-1 with the detailed DDP model are compared below to the ideal pump model:

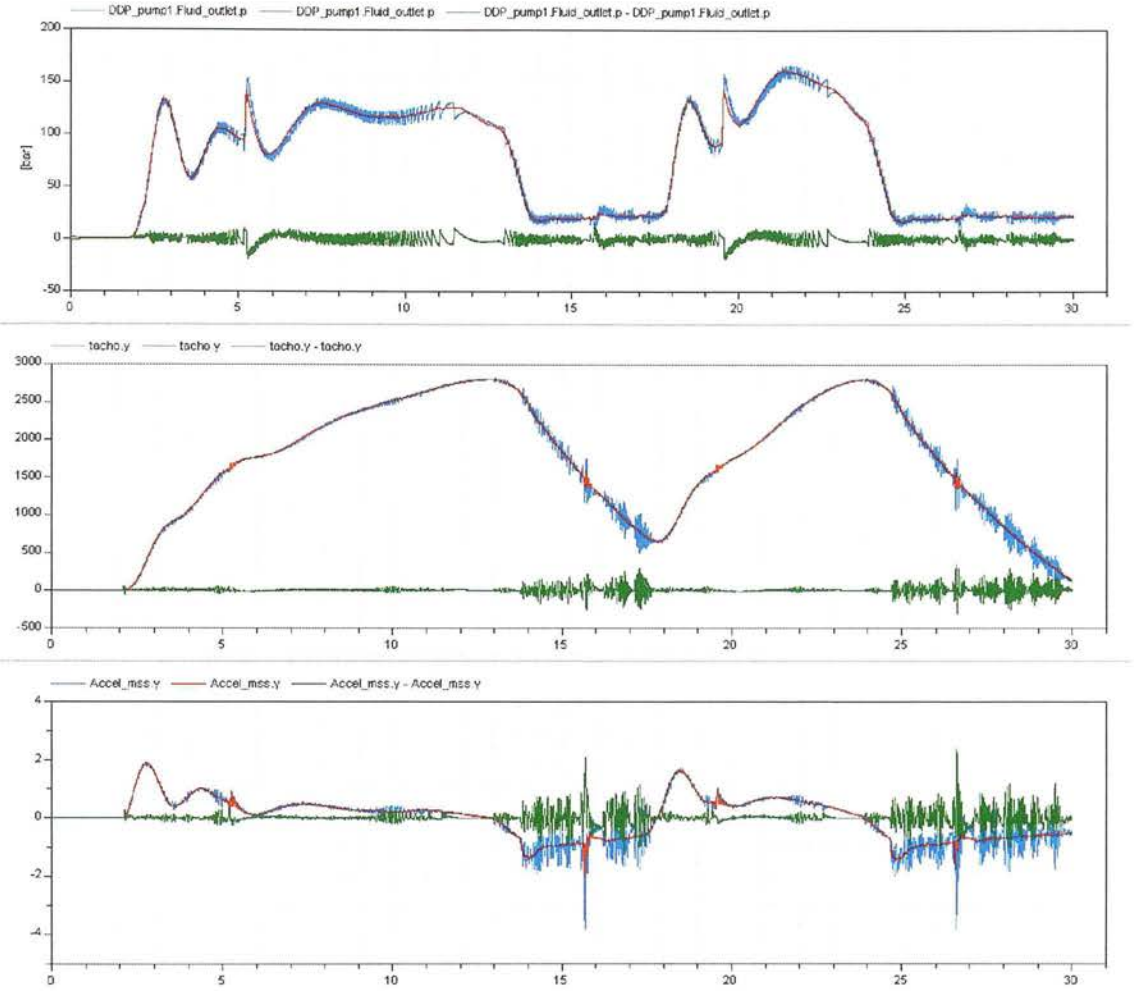


Figure 6-37: Test 1-1, comparison between simulation results with detailed DDP model (red), ideal pump model (blue) and the difference (green); top trace=pump pressure; middle trace=motor speed; bottom trace=vehicle acceleration

The results above show that the effect of the DDP flow pulsation is to introduce a pressure pulsation of approximate amplitude +/-10 bar. During deceleration the DDP pulsation causes a substantial motor speed oscillation, which is reflected in the overall vehicle acceleration. The combination of the overcentre valve and the driveline compliance appears to magnify the effect of the DDP pulsation on the motor speed, although these high-frequency motor speed oscillations were mostly absorbed in the driveline compliance

The experimental results for test 7-1 are compared below with the ideal pump simulation and the detailed DDP simulation. The "motor speed" traces are the speed of the vehicle from the motor speed (measured with the tacho in the experiment), while the "vehicle speed" traces are the actual speed of the vehicle (measured by differentiating the position sensor signal in the experiment).

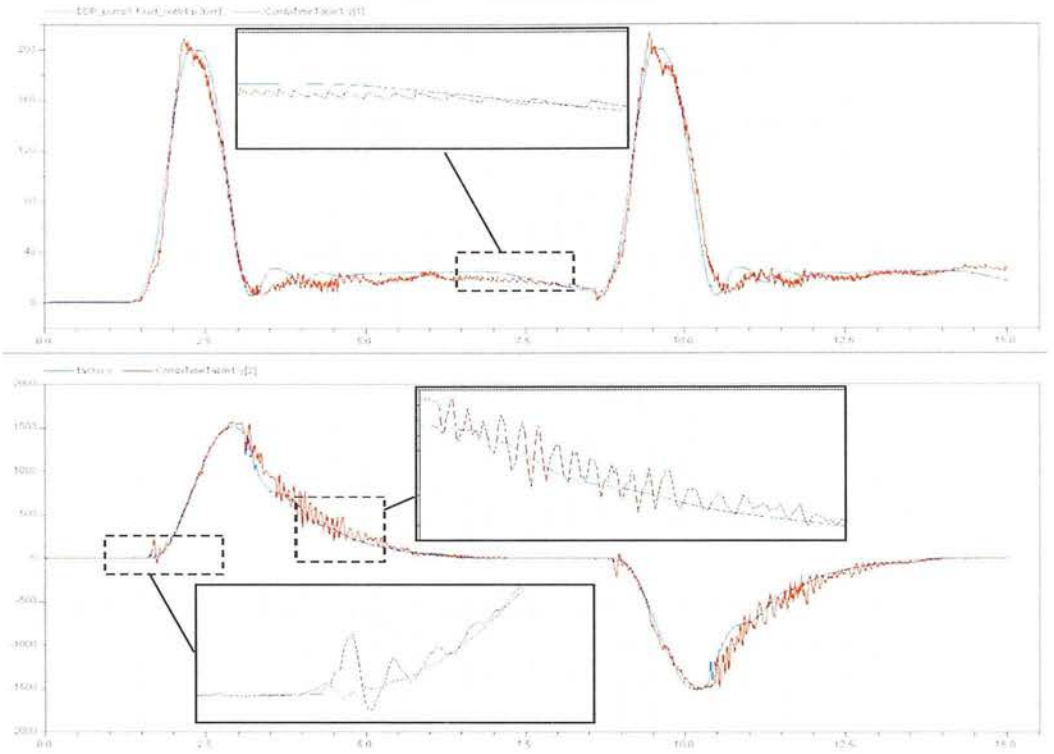


Figure 6-38: Test 7-1, ideal pump simulation (blue) and experiment (red); top trace= pressure, bottom trace=motor speed

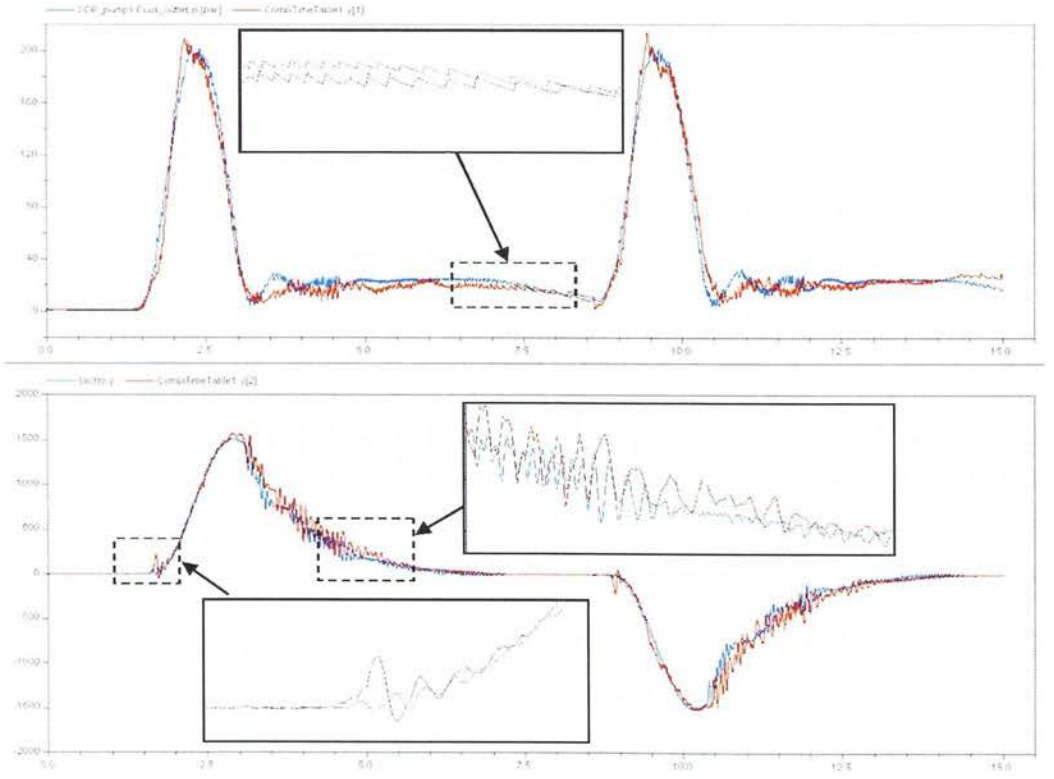


Figure 6-39 Test 7-1, detailed DDP simulation (blue) and experiment (red); top trace= pressure, bottom trace=motor speed

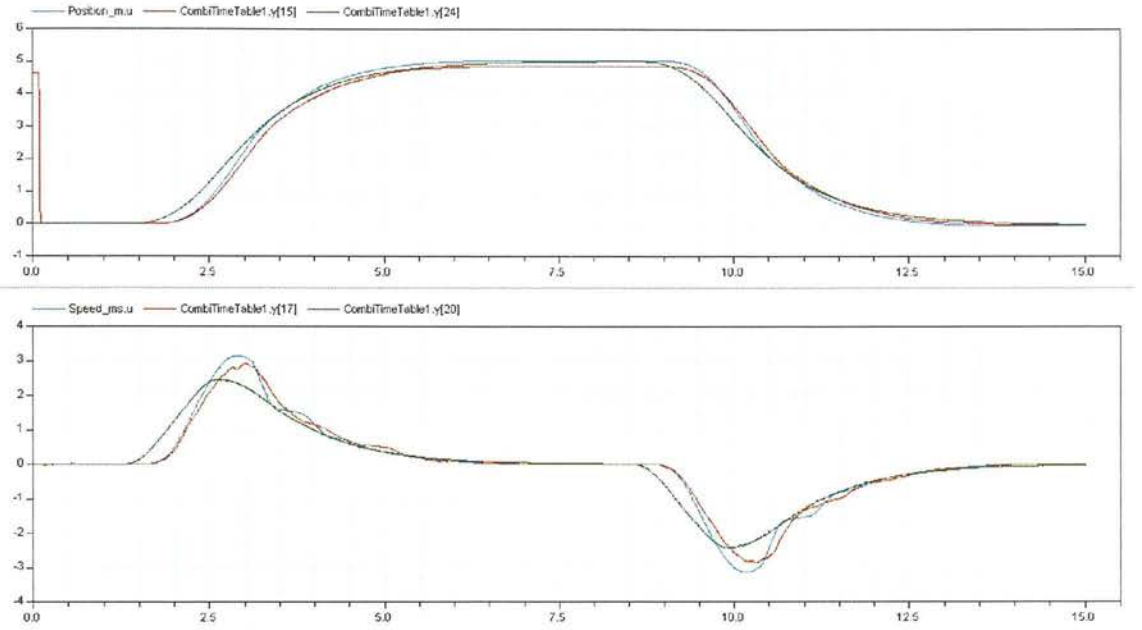


Figure 6-40: Test 7-1, ideal pump simulation (blue), experiment (red), expected from displacement demand (green); top =vehicle position, bottom =vehicle speed

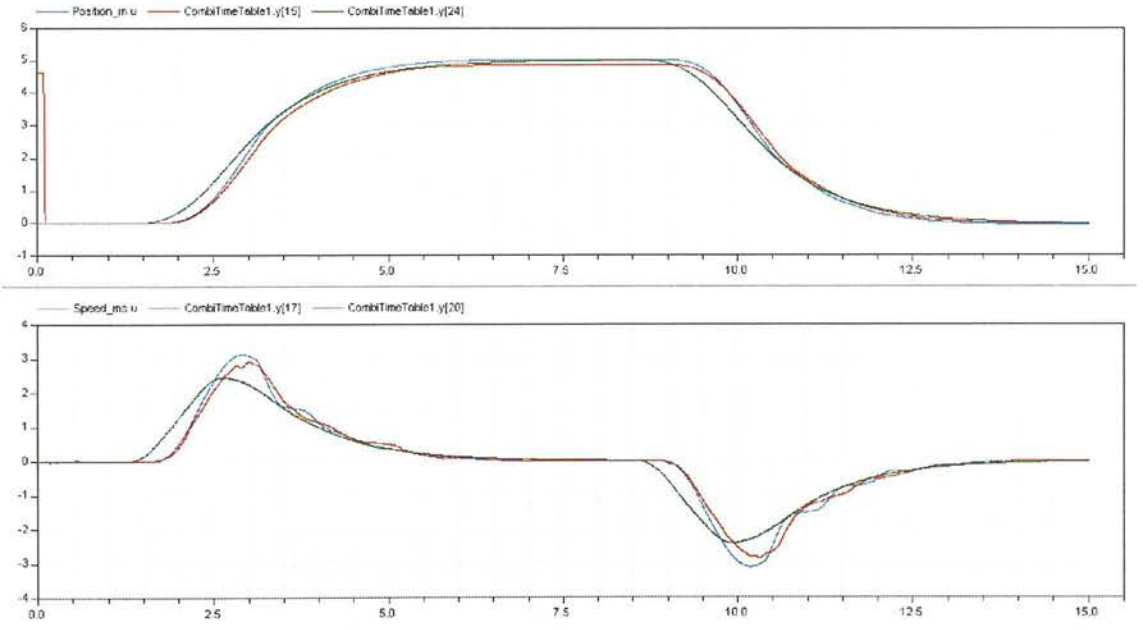


Figure 6-41: Test 7-1, detailed DDP simulation (blue), experiment (red), expected from displacement demand (green) ; top=vehicle position, bottom =vehicle speed

Clearly the ideal pump simulation gives a fairly good approximation to the overall low-frequency behaviour. There is hardly any difference in the speed and position traces from the ideal pump and the detailed DDP simulations.

The high-frequency motor speed oscillations under braking, visible in the DDP simulation and the experimental results, are completely absorbed in the driveline compliance; the simulated and experimental vehicle speed do not show oscillations.

The detailed DDP model displays similar high-frequency behaviour to the experimental results. However, it can be concluded that for much simulation of overall vehicle behaviour, an ideal pump approximation is sufficient.

6.7.3 Comparison of DDP flow algorithms

The above results show that the detailed DDP simulation matches the experimental data fairly accurately, both in terms of the hydraulic/mechanical system dynamics and the effect of the DDP flow pulsation. Once this validation was achieved, the simulation model was used to investigate the effect of the choice of flow control algorithm on the behaviour of the system. Three algorithms were considered, namely:

- The *full stroke algorithm* mixes full and idle strokes.
- The *fixed part stroke algorithm* mixes partial strokes (0.166) with idle strokes below a displacement demand=0.166, above which it transitions to mixing full strokes and idle strokes.
- The *variable part stroke algorithm* enables every cylinder all the time, with modulation of flow coming from changing the timing of the LPV actuation.

These algorithms were compared using the demand recorded during Test 7-1; this is a sequence of position steps in the 'slow' and 'loose' control mode; the effect of DDP pulsation is most obvious when the driver demand is slow and smooth.

First, the simulated flow output of the DDP is compared for these three algorithms:

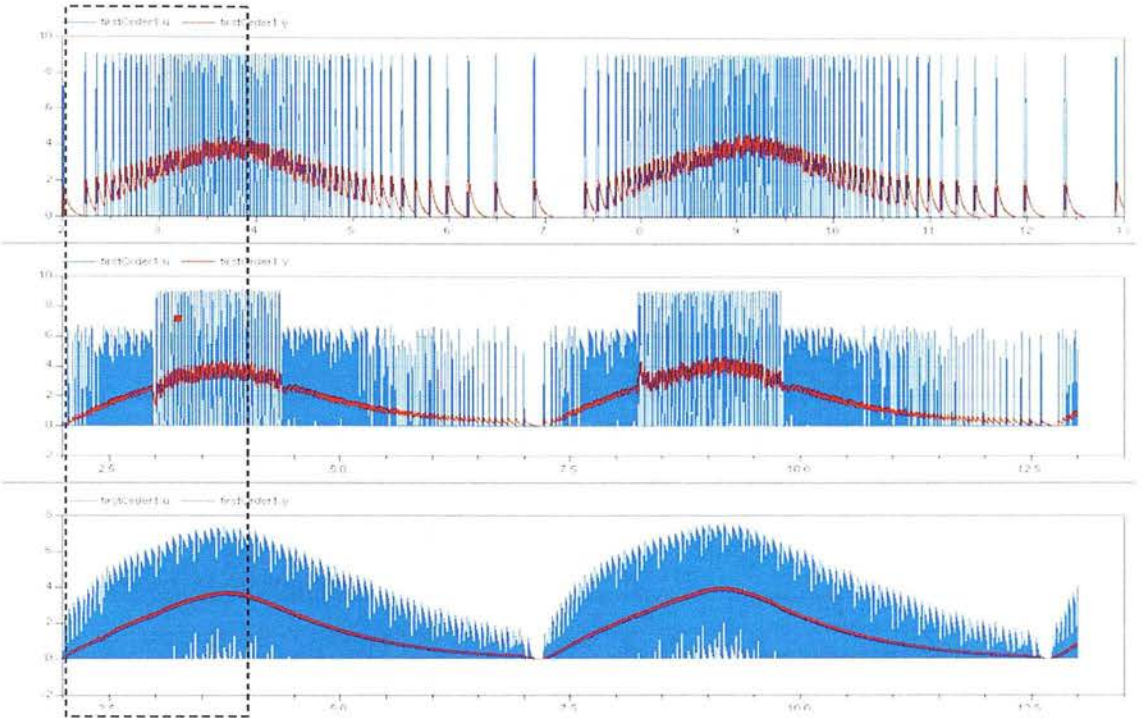


Figure 6-42: Simulated DDP flow for Test 7-1 (blue=flow, red=flow filtered with first order low-pass τ=0.05s); top=full strokes, middle= fixed part stroke (ps_fraction=0.166), bottom=variable part stroke

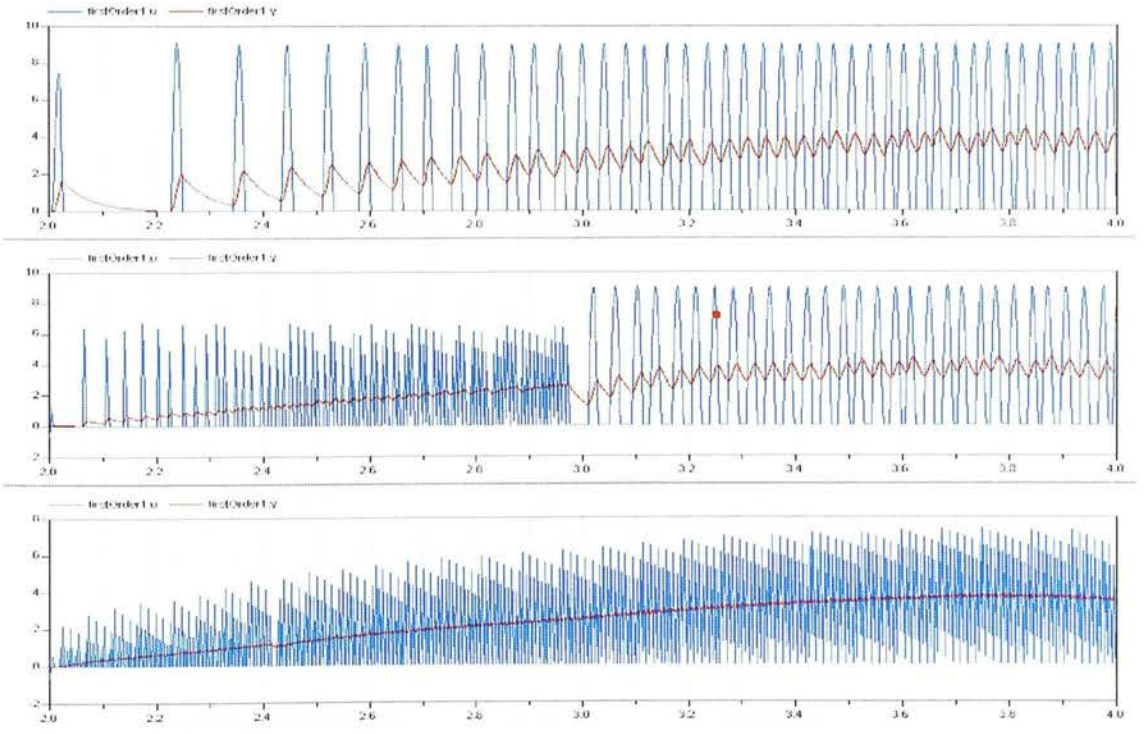


Figure 6-43: Zoom in on t=2s to t=4s of figure above

The simulated response of the system to the flow pulsations above are shown below:

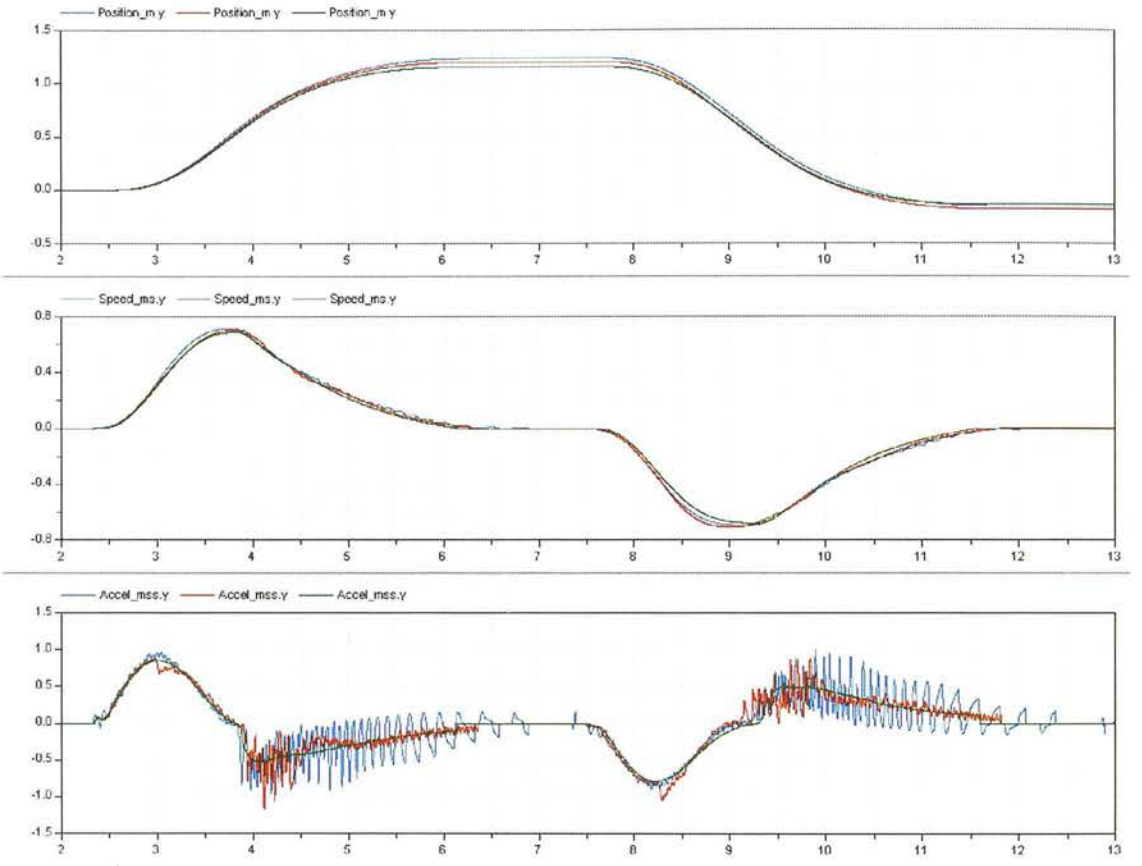


Figure 6-44: Test 7-1; simulation of the effect of the DDP flow algorithm: blue=full strokes, red=fixed part stroke (ps_fraction=0.166), green=variable part stroke; top=position, middle=speed, bottom=acceleration

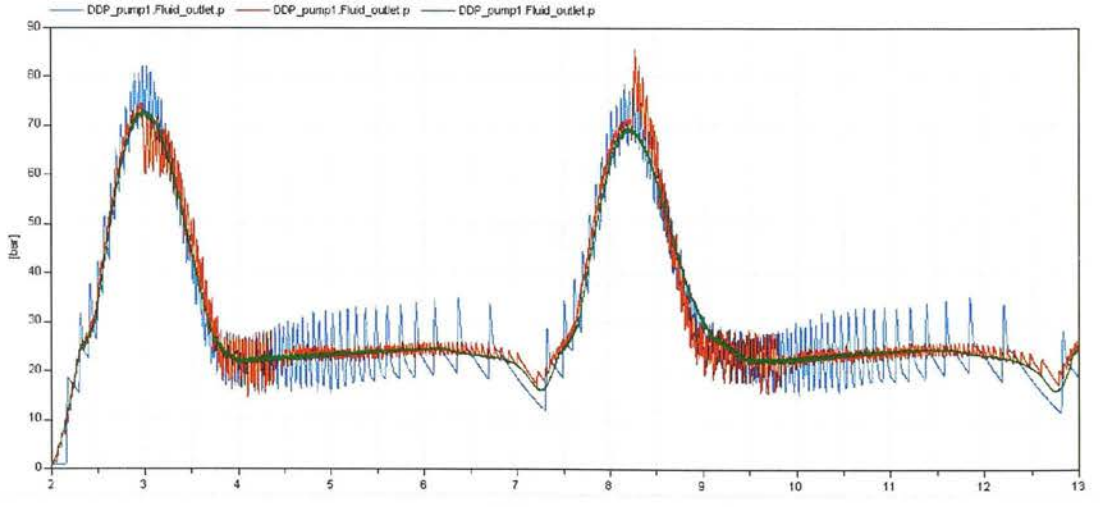


Figure 6-45: Pressure trace for above tests

The results above show that the position trace is similar for all of the algorithms. However, the full stroke algorithm produces ripples in the vehicle speed and acceleration traces; close to zero speed the acceleration becomes discontinuous and the vehicle moves in discrete steps; this behaviour is not seen in results from the fixed or variable part stroke algorithms. This is similar to the experimental results from tests 9-4 and 9-5 presented in Section 5.4.4.

The pressure trace shows that the fixed part stroke algorithm produces large pressure pulsations at all displacement demands; as expected, similar pulsations are seen from the fixed part stroke algorithm with displacement demand above 0.166. Below this displacement demand, the fixed part stroke algorithm produces a fairly smooth pressure trace. The variable part stroke algorithm produces the smoothest pressure trace at all displacement demands.

6.7.4 Effect of system compliance

The magnitude of the pressure pulsation caused by the DDP flow control algorithm is fundamentally related to the amount of system compliance. It was therefore investigated whether it possible to achieve smooth control of the vehicle with the full stroke algorithm, by simply increasing system compliance.

The simulation for test 7-1 was run with a range of values for the "compliance number" CN, by changing the volume of the closed fluid volume, which models the overall system compliance:

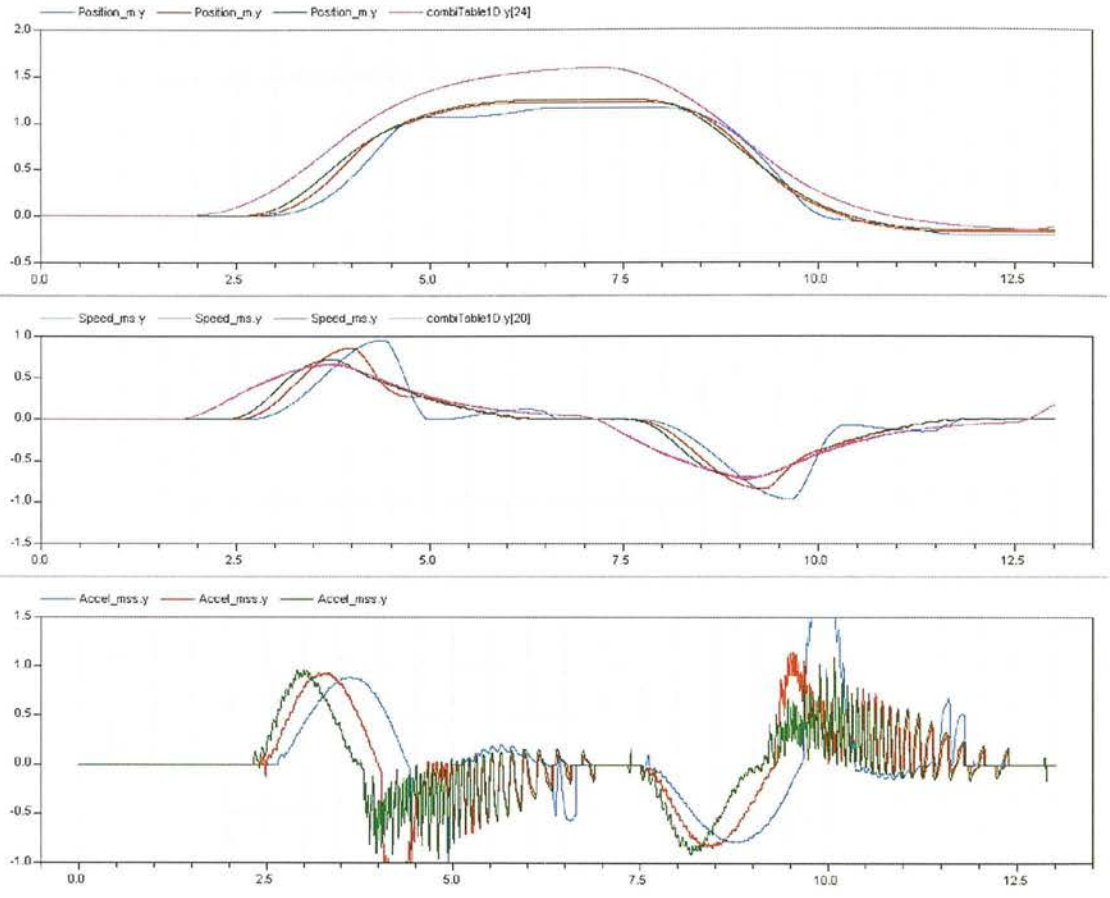


Figure 6-46: Effect of system compliance on vehicle behaviour with full strokes; blue:v=6.6l(CN=43.2), red:v=3.3l(CN=21.6); green:v=1.65l(CN=10.8); pink=ideal behaviour from displacement demand; top trace=position, middle trace=velocity, bottom trace=acceleration

The pressure trace shows the expected effect on pressure ripple: higher compliance leads to lower pressure ripple. Even with the highest compliance, the vehicle moves in discrete steps at low speed as seen from the acceleration trace (blue). The velocity and position traces show that as CN increases, the overall vehicle behaviour becomes unstable. As the fundamental frequency of the system reduces there is insufficient damping of the overcentre valve oscillations. This result indicates that for the vehicle studied, the fixed part stroke algorithm cannot be expected to produce smooth vehicle control; low compliance leads to unacceptable pulsation, while high compliance leads to instability when braking.

It should be noted that all these results show the case of a 6-cylinder DDP. Smoother pressure pulsation would of course result from using more cylinders of smaller displacement. Seven cylinders may be the practical limit for a single bank in the radial piston configuration; any number above that would seem to require multiple banks, causing a significant increase in production cost.

6.7.5 Comparison with swashplate pump

Axial-piston swashplate pumps have the following non-ideal features which are particularly relevant when considering low-speed vehicle behaviour:

- The volumetric losses at low swash angles are very similar to that at full swash (Dorey 1988); the volumetric losses in the DDP are proportional to the displacement fraction.
- The swash angle is controlled by an electrohydraulic servomechanism; deadband and hysteresis become very significant at low swash angles; by contrast the DDP has zero deadband and hysteresis.

The response speed of the servo is not a significant effect at low speeds with gentle demand signals; typically the swash angle can ramp from zero to maximum in 300ms (Akers and Lin 1987).

To investigate the implications of these features, the ideal pump model was modified to simulate a swashplate pump with electronic displacement control (EDC). It should be noted that swashplate pumps are more usually applied to closed-circuit transmissions without overcentre valves, therefore the comparisons here are somewhat artificial. However, such a comparison does yield useful insight into effect of the non-ideal features on swashplate pump performance, and is to some extent relevant to the closed circuit case.

The effect of volumetric losses in the swashplate pump is dealt with by adding a laminar impedance of 0.045l/min per bar, between pump outlet and tank. This was chosen to be equivalent to the leakage in a 12cc pump which exhibits 95% volumetric efficiency at 200 bar and 1500rpm, giving a leakage of 0.9l/min at 200 bar.

The displacement fraction demand is processed through a deadband of 2% and a hysteresis band of 5% – these values are typical for production swashplate pumps with electronic displacement control (Sauer-Danfoss datasheet MCV105V 1999). In fact the control deadband is typically much larger (>10%) but much of this is constant and can therefore be compensated by the electronic controller. However, it is always necessary to have some deadband, to ensure that the vehicle does not creep with zero demand - it is assumed here that there is 2% "residual deadband" after compensation in the controller.

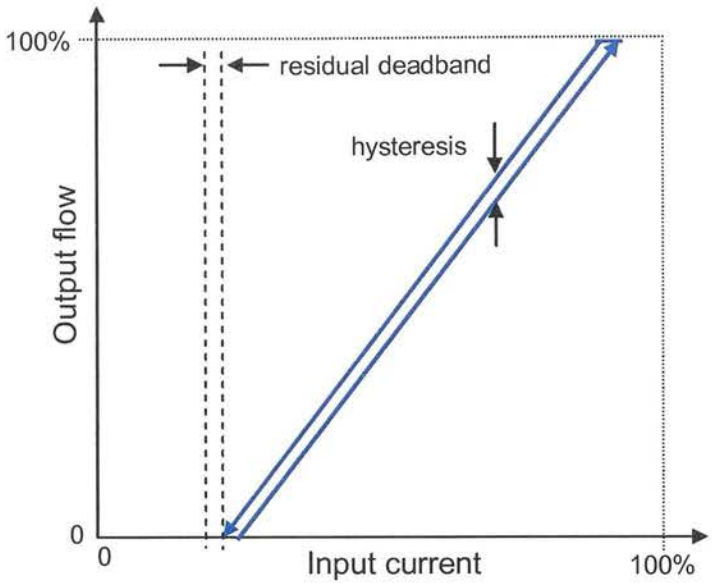


Figure 6-47: Hysteresis and deadband in a swashplate pump with EDC

The results from the simulation are shown below, compared with actual and simulated DDP results (fixed part stroke =0.166), and ideal traces from the displacement demand signal:

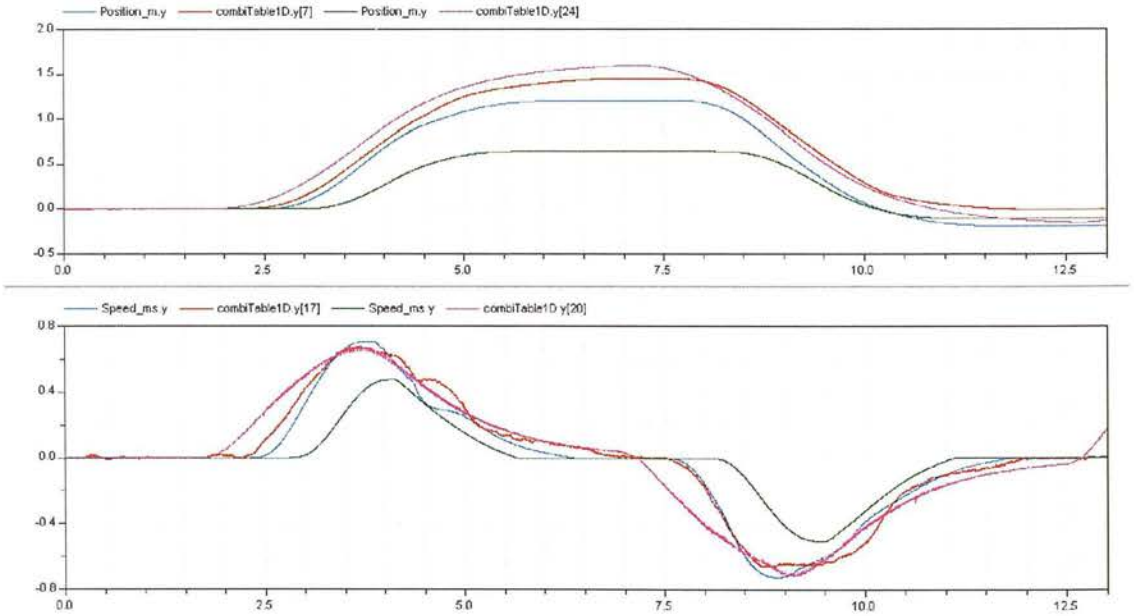


Figure 6-48: Test 7-1; comparison of swashplate pump with simulated and experimental DDP; blue=simulated DDP, red=experiment with DDP, green=simulated swashplate pump, pink=expected behaviour from and ideal system from displacement demand; top trace=position, bottom trace=speed

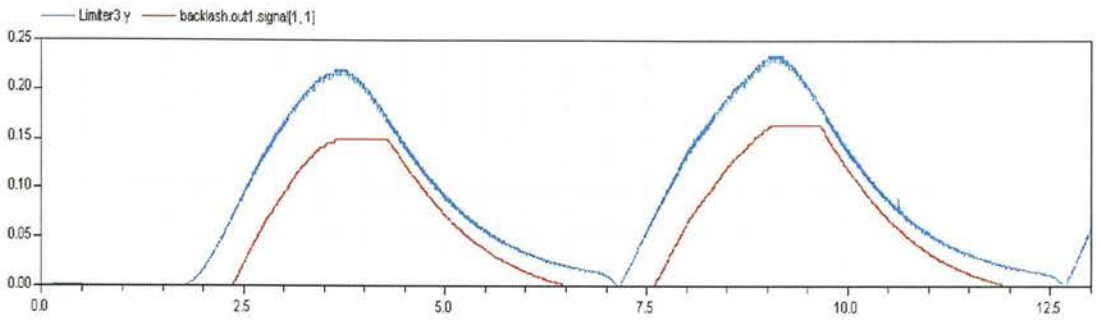


Figure 6-49: Displacement demand and actual swashplate displacement in above simulation

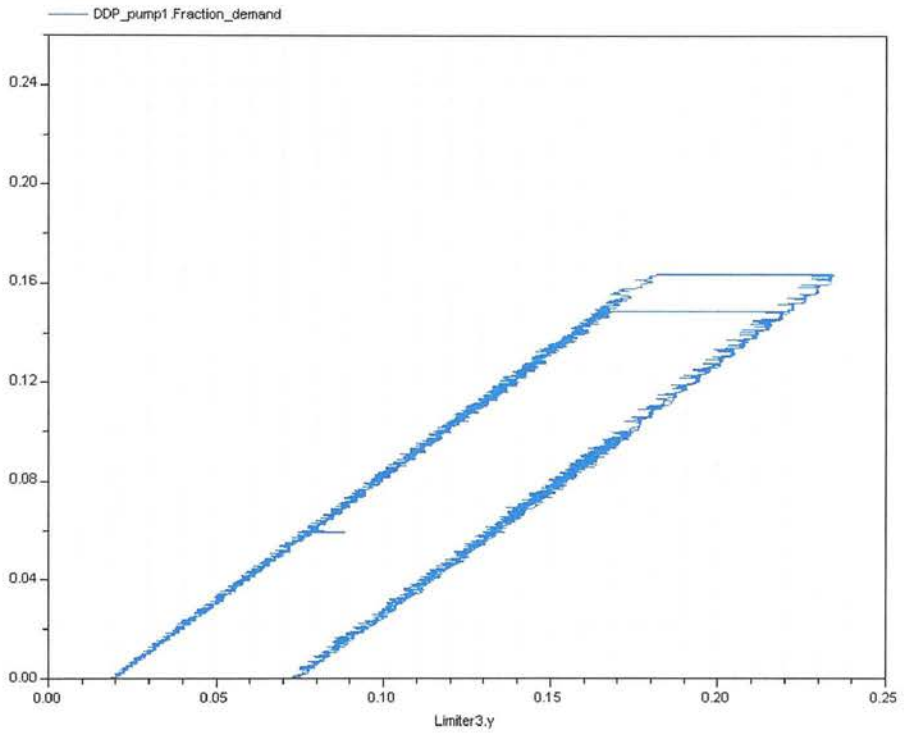


Figure 6-50: Simulated swashplate hysteresis loop in above simulation; x-axis=demand, y-axis=actual displacement fraction

The hysteresis loop from the swashplate simulation clearly shows the 2% residual deadband and 5% hysteresis band.

These deadband and hysteresis effects mean that the swashplate pump starts to deliver flow approximately 0.5s after the displacement demand starts to rise at t=1.75s. There is a further 0.6s delay before the vehicle starts to move; the greater leakage in the swashplate pump means that it takes longer to raise sufficient pressure to overcome motor and tyre stiction. There is a total delay between demand and movement of 1.1s. By contrast the experimental

results with the DDP show that the vehicle started to move 0.5s after the demand started to rise.

There is a similar effect when decelerating to a stop. With the swashplate pump, the vehicle stops abruptly at t=5.5s even though the demand continues to gently tail-off and does not reach zero. The experimental results with the DDP show that the vehicle continued to move until t=7s, when the directional valve changed over.

During deceleration, there is less instability of the overcentre valve with the simulated swashplate pump than with either the simulated or experimental DDP results; this is perhaps to be expected, because higher pump leakage should increase damping of the system fundamental frequency.

Looking at the position trace, the swashplate pump achieves only half the commanded position, whereas the experimental results with the DDP show it achieved 90% of the commanded position. Clearly open-loop position control is much more accurate with the DDP.

It should be noted that the electronic controller for the swashplate pump could also compensate for some of the hysteresis band, but this may not be constant with respect to ramp rate, temperature and swashplate moments due to pressure, and it is therefore impossible to eliminate without feedback of the actual swash angle to the controller. Electronic swashplate position feedback loops with an integral term are used in swashplate machines with the highest demands of control linearity, but the added cost and complexity seem to rule it out of most mobile applications.

6.8 Conclusions

- Non-linear time-domain modelling is an effective means of simulating the behaviour of an open-circuit DDP propel system with an overcentre valve; the Dymola software used is well-suited to the task. Such a model has been created and validated with experimental results.
- In designing such a system, the size of the hydraulic compliance must be carefully considered to achieve acceptable performance, especially with regard to braking with the overcentre valve.
- Modelling the DDP as an ideal continuous pump produces accurate results for low-frequency system behaviour, but the detailed DDP model described is necessary to model the high-frequency pulsation observed in experiments.

- The accurate control achievable with the DDP at very low displacement has been shown to provide significant advantages over the swashplate pump for open-loop position control at very low vehicle speed.
- It is unlikely that acceptable pulsation at low speed can be achieved in vehicle propel applications without some form of part-stroke flow algorithm.
- Dead volume in the DDP cylinders causes the amount of flow produced from small part strokes to be strongly sensitive to pressure, but this does not rule out the use of part strokes in general.
- The fixed part-stroke flow algorithm can provide smooth motion control at low speed.
- The variable part-stroke algorithm seems to promise the smoothest control of vehicle speed and the lowest magnitude of pressure ripple, but at the expense of an increase of high-frequency pressure pulsation, mechanical stress, internal pump energy losses and noise.

7 Conclusions

As was stated in the introduction, this work is essentially a study of the feasibility of applying the DDP and the DDPM to hydrostatic transmission systems for propelling a vehicle. Many different aspects of the subject have been studied, at system, component and sub-component levels. Overall, this work has demonstrated that the DDP and the DDPM can be successfully applied to such systems. Detailed conclusions have been presented at the end of each chapter, and these will not be repeated here. However, referring to the thesis in Section 1.3, this work has conclusively shown that:

1. The efficiency of the DDP can be well described with a semi-empirical mathematical model.
2. The DDP offers very significant energy efficiency advantages compared to the swashplate pump.
3. The performance of the DDPM is crucially dependent on optimising the response speed of the commutating valves, and magnetic FEA is an effective tool to achieve this goal.
4. A DDPM can be made which can start from zero speed, and operate with variable displacement in all four quadrants.
5. A hydrostatic transmission comprising a DDP and a DDPM is capable of propelling a vehicle.
6. A hydrostatic transmission comprising a DDP and a conventional motor can be constructed which exhibits smooth motion control without perceptible pulsation.
7. The DDP offers control advantages compared with the swashplate pump for vehicle propel transmissions.
8. Non-linear time-domain simulation is an effective tool for analysing such systems.

7.1 Recommendations for further work

Further work to take the subject matter forward could include the following:

- The efficiency analysis presented here is focussed on the DDP but further study is needed relating to the DDPM efficiency.
- An open loop transmission comprising a DDP and conventional motors relies upon external valves to achieve four-quadrant control; the effect of the losses in these valves on the overall system efficiency deserves further investigation.
- It has been shown that magnetic finite element analysis is an effective tool for improvement of the performance of the valve solenoids. The author found that fully automatic optimisation was not a appropriate approach due to the difficulty of encoding practical constraints outside of the domain of magnetics. The application of expert system methods may allow a definitively optimum design to be achieved.
- While a DDPM capable of propelling a vehicle has been demonstrated, the 'false rev' method of starting has some disadvantages. It is worth investigating whether these can be ameliorated by control system improvements or whether a fundamentally different method is required.
- The DDP of radial piston design has been shown to be capable of high efficiency, but there is still potential for further improvements. In the author's view these are mainly to come from reducing idling losses, perhaps by reducing off-loaded piston pad losses or by reducing the pressure-drop through the disabled LPV.
- The fixed part stroke algorithm is only one of many possible methods for controlling DDP displacement and further computer simulation studies may reveal an optimum method.

8 References

Akers A, Lin J, Control of an axial piston pump using a single-stage electrohydraulic servovalve, Proceedings of the American Control Conference, Minneapolis MN (USA), pp. 1865-1870, 1987

Aldefeld B, A numerical solution of transient nonlinear eddy-current problems including moving iron parts, IEEE Transactions on Magnetics, Vol. Mag-14, No. 5, pp. 371-373, September 1978

AMESIM S.A., Amesim 4.3.1, Roanne (France), 2005,

http://www.amesim.com/software/libraries/electromechanical.aspx

Andersen TO, Hansen MR, Pedersen P, Conrad F, The influence of flow force characteristics on the performance of overcentre valve systems, SICFP05, Linkoping (Sweden), June 1-3 2005

ANSYS Inc., Ansys 5.4 Analysis Guide, Canonsburg PA (USA), 1997, http://www.ansys.com

Åström KJ, Elmqvist H, Mattsson SE, Evolution of continuous-time modeling and simulation, Twelfth European Simulation Multiconference, ESM '98, Manchester (UK), June 16-19 1998

Bauer G, Ölhydraulik (8th Edition), 2005, Teubner BG Gmbh

Bavendiek R, Verlustkenwertbistimmung am Beispiel von hydrostatischen Maschinen in Schragachsenbauweise; VDI Fortschrittberichte Reihi 7 Nr. 122, VDI Verlag, 1987

Beater P, Modelling and digital simulation of hydraulic systems in design and engineering education using Modelica and HyLib, Proceedings of Modelica Workshop 2000, Lund (Sweden), pp.33-40, Oct 23- 24 2000

Beater P, Modelling of hydraulic systems; Tutorial for HyLib® 2.3.25, 2005, Dynasim AB (Sweden)

Beater P, Otter M, Multi-domain simulation: mechanics and hydraulics of an excavator, Modelica 2003, pp. 331-341, November 3-4 2003, Linkoping (Sweden)

Bowns D, McCandlish D, Edge K, Some considerations in optimisation of efficiency of a hydrostatic transmission and engine combination, First European Fluid Power Conf., East Kilbride (UK), Paper No. 37,10-12 September 1973

Canbulut F, Sinanoglu C, Yildirim S, Neural network analysis of leakage oil quantity in the design of partially hydrostatic slipper bearings, Industrial Lubrication and Tribology, Vol 56 No. 4, pp. 231-243, 2004

Chapple PJ, Displacement control in hydrostatic motors, I Mech E Conference on Hydrostatic Transmissions for Vehicle Application, Warwick (UK), Paper C140/81, pp. 1-7, 29-30 September 1981

Chapple PJ, Modelling of a radial-piston hydraulic motor, Proc. Instn. Mech. Engrs., Vol 206, pp.171-180, 1992

Cheung NC, Lim KW, Rahman MF, Modelling a linear and limited travel solenoid, Proceedings of the IECON '93 Intnl. Conf. on Industrial Electronics Control and Instrumentation, Maui HI (USA), vol. 3, pp. 1567-1572, 15-19 November 1993

Chladny RR, Koch CR, Modelling automotive gas-exchange solenoid valve actuators, IEEE Transactions on Magnetics, Vol 41, No. 3, pp. 1155-1162, March 2005

Conrad F, Sorensen PM, Trostmann E, On the modelling of flow- and torque-loss in hydrostatic machines, Proc. 9th Intl. Symp. on Fluid Power, BHR Group, STI, Cambridge, pp. 3-17, April 1990

Conway JT, Exact solutions for the magnetic fields of axisymmetric solenoids and current distributions, IEEE Transactions on Magnetics, Vol 37, No. 4, pp. 2977-2988, July 2001

DeCarlo RA, Zak SH, Mathews GP, "Variable structure control of nonlinear multivariable systems: A tutorial", Proc. IEEE, Vol 76, No. 3, pp. 212-232, March 1988

Dorey RE, Modelling of losses in pumps and motors, First Bath International Fluid Power Workshop: Design, Modelling and Control of Pumps, Bath (UK), pp.71-97, 8 September 1988

Douglas JF, Gasiorek JM, Swaffield JA, Fluid Mechanics (3rd edition), Longman Scientific & Technical, 1995

Eaton Corp., Vickers Axial Piston Pump Catalog GB2379B, 2006, www.eaton.com

Edge KA, Brett PN, Leahy JC, Digital computer simulation as an aid in improving the performance of positive displacement pumps with self-acting valves, Proceedings of the Institute of Mechanical Engineers, Vol 198, No. 14, pp. 267-274, 1984

Edge KA, Brett PN, The pumping dynamics of a positive displacement pump employing self-acting valves, Transactions of the ASME. Journal of Dynamic Systems Measurement and Control, Vol 112, pp. 748-754, Dec 1990

Ehsan Md, Rampen W.H.S and Salter S.H, "Computer Simulation of the Performance of Digital-Displacement Pump-Motors", Proceedings of ASME International Mechanical Engineering Congress and Exposition, FPST Volume-3, pp.19-24, Atlanta Nov. 1996

Ehsan Md, Rampen W.H.S and Salter S.H, "Modelling of Digital-displacement Pump-Motors and their Application as Hydraulic drives for Non-uniform Loads", Journal of Dynamic Systems, Measurement and Control, Transactions of ASME, Vol.122, pp.210-215, March 2000.

Ehsan Md, Rampen W.H.S and Taylor J.R.M, "Simulation and Dynamic Response of Computer Controlled Digital Hydraulic Pump/Motor System Used in Wave Energy Power Conversion", Proceedings of 2nd European Wave Power Conference, Lisbon, pp. 305-311, 8-10 Nov. 1995

Ehsan Md., "A New Type of Malone Engine with Digital-Displacement Hydraulic Drive", Ph.D. Thesis, University of Edinburgh, Scotland, UK, June 1997.

Ehsan Md., Rampen W.H.S, "Using Digital-Displacement Hydraulic Drive for Controlling Piston Motion of Heat Engines", TH-11/1, Proceedings of 2nd International Conference on Fluid Mechanics and Fluid Power, IIT-Roorkee, India, pp. 993-1000, December 12-14, 2002

Ewing J, Siemens: If at first you don't succeed..., Business Week Online, MacGraw-Hill, September 5 2005,
http://www.businessweek.com/magazine/content/05_36/b3949077_mz054.htm

Fairchild Semiconductor, PSPICE model for 2N3390 transistor, Maine USA, 1999, http://www.fairchildsemi.com/pf/2N/2N3390.html

Harris RM, Edge KA, Tilley DG, Predicting the behaviour of slipper pads in swashplate-type axial piston pumps, Journal of Dynamic Systems Measurement and Control, Vol 118, pp.41-47, 1996

Hayashi S, Nonlinear phenomena in hydraulic systems, Paper 1-3, ICFP2001, 3-5 April 2001, Hangzhou (China)

Heggie WS, Davies AS, The development and testing of a regenerative diesel-hydraulic power train for the urban bus, 7th Intl. Fluid Power Symposium, Bath (UK), Paper 40, p351-357, 16-18 September 1986

Hooke CJ, Li KY, The lubrication of centrally-loaded slippers in axial piston pumps-centrally loaded behaviour, Proc. Instn. Mech. Engrs. Vol 202 No. C4, pp.287-293, 1988

Huhtala K, Karen JH, Vilenius MJ, Semi-empirical modelling of losses in components of hydrostatic transmission systems, 8th Bath Intl. Fluid Power Workshop, Bath (UK), pp. 237-253, 20-22 September 1995

Huhtala K, Vilenius MJ, Comparison of steady-state models of hydraulic pump, Proc 5th Scandinavian Int. Conf. on Fluid Power (SICFP '97), Vol 3, Linkoping (Sweden), pp. 139-155, 28-30 May 1997

Jen, YM, Lee CB, Influence of an accumulator on the performance of a hydrostatic drive with control of the secondary unit, Proceedings of the Institution of Mechanical Engineers. Part I, Journal of Systems & Control Engineering, v 207, n. 3, pp. 173-184, 1993

Johnston DN, Numerical modelling of reciprocating pumps with self-acting valves, Proceedings of the Institute of Mechanical Engineers, Vol 205, pp. 87-96, 1991

Kajima T, Nakamura Y, Sonoda K, Development of high speed solenoid valve-investigation of the energizing circuits, Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation, pp.564-569, 9-13 Nov 1992

Kawase Y, Ohdachi Y, Dynamic analysis of automotive solenoid valve using finite element method, IEEE Transactions on Magnetics, Vol 27, No. 5, pp. 3939-3942, September 1991

Kim CS, Lee CO, Speed control of an overcentred variable-displacement hydraulic motor with a load-torque observer, Control Engineering Practise, Vol. 4, No.11, pp.1563-1570, Nov 1996

Koc E, Hooke CJ, Considerations in the design of partially hydrostatic slipper bearings, Tribology International, Vol 30 No. 11, pp. 815-823, 1997

Koc E, Hooke CJ, Investigation into the effects of orifice size, offset and overclamp ratio on the lubrication of slipper bearings, Tribology International, Vol. 29 No. 4, pp. 299-305, 1996

Koc E, Hooke CJ, Li KY, Slipper balance in axial piston pumps and motors, Trans. ASME, Vol 114, pp.766-772, October 1992

Komatsu America Corp., Komatsu Introduces WA320-5 Wheel Loader With Hydrostatic Transmission, Press release, December 1 2003, http://www.komatsuamerica.com/index.cfm?resource_id=2032

Krus P, Modelling and analysis of the dynamic properties of mobile hydraulic systems, 4th Bath Intnl. Fluid Power Workshop (Fluid Power Systems and Modelling), Bath, pp. 327-345, Sept. 1991

Johnston DN, Numerical modelling of reciprocating pumps with self-acting valves, Proc. Instn. Mech. Engnrs., Vol 205, pp.87-96, 1991

Lennevi J, Palmberg J-O, The signficance for control design of the engine and power transmission coupling in hydrostatic drivetrains, 8th Bath International Fluid Power Workshop, Bath (UK), pp.41-59, 20-22 Sept. 1995

Lequesne B, Dynamic model of solenoids under impact excitation including motion and eddy current effects, IEEE Transactions on Magnetics, Vol 26, No. 2, pp. 1107-1116, March 1990

Lewandoski E, Benson D, Fronczak F, Beachley N, Test rig design for hydrostatic accumulator energy-storage automobile research, American Society of Mechanical Engineers Design Engineering Division (Publication) DE, v 11, pp. 45-52, 1987

Liebenberg L, Kruger JJ, Computer controlled hydrostatic transmission with traction control for vehicles, SAE Transactions, v 99, n. Sect 2, pp. 306-317 (SAE 901559), 1990

Lynn A, Smid E, Eshraghi M, Caldwell N, Woody D, Modelling hydraulic regenerative hybrid vehicles using AMESim and Matlab/Simulink, Proceedings of SPIE - Volume 5805 Enabling Technologies for Simulation Science IX, pp. 24-40, May 2005

Manring ND, Wray CL, Dong Z, Experimental studies on the performance of slipper bearings within axial-piston pumps, ASME Journal of Tribology, Vol 126, pp. 511-518, July 2004

Mathew J, Hippner M, Optimisation of finite element aided design of electromagnetic devices, IEEE AFRICON Conference, Gaborone (Botswana), Vol. 2, pp. 651-654, September 15-17 2004

McCandlish D, Dorey RE, The mathematical modelling of hydrostatic pumps and motors, Proceedings of the Institution of Mechanical Engineers, Vol 198B, No. 10, pp. 165-174, 1984

Meeker DC , Finite Element Method Magnetics, Version 4.0.1 (10 April 2006 Build), http://femm.foster-miller.net

Mikeska D, Ivantysynova M, Virtual prototyping of power split drives, Proc. Bath Workshop on Power Transmission and Motion Control PTMC 2002, Bath (UK), pp. 95-111, 2002

Myatt RL, Pillsbury RD, Electromagnetic analyses of the GEM detector magnet system for SSC, IEEE Transactions on Magnetics, Vol 30, No. 5, pp. 270-273, September 1994

Nguyen-Shaefer H, Flow induced noise in a hydraulic plunger pump, Proc. Bath Workshop on Power Transmission and Motion Control PTMC 1998, Bath (UK), pp.219-231, 1998

Nikolaus H, Hydrostatic mobile and winch drives with energy reclamation, I Mech E Conference on Hydrostatic Transmissions for Vehicle Application, Warwick (UK), Paper C146/81, pp. 55-58, 29-30 September1981

Ohdachi Y, KawaseY, Murakami Y, Inaguma Y, Optimum design of dynamic response in automotive solenoid valve, IEEE Transactions on Magnetics, Vol. 27, No. 6, Part 2, pp. 5226-5228, November 1991

Okoshi T, Resistance network analogue simulation of the magnetic field produced by a solenoid, IEEE Transactions on Electron Devices, Vol. ED-12, No.10, pp.564-573, October 1965

Ortwig H, Error reduced simulation of hydrostatic transmission performance, The Eighth Scandinavian Intl. Conf. on Fluid Power SICFP '03, Tampere (Finland), May 7-9 2003

Overdiek G, Design and characteristics of hydraulic winch controls by counterbalance valves, IMechE Conference - Hydrostatic transmissions for vehicle application, Warwick (UK), pp.39-48, 29-30 September 1981

Paoluzzi R, Zarotti LG, Ruggeri M, Fantuzzi C, Electronic Hybrid Limiter (EHL) for pump displacement, Proceedings of ASME International Mechanical Engineering Congress and Exposition, FPST Volume-3, pp.25-32, Atlanta Nov. 1996.

Parker-Hannifin Corp., Series P3 Catalog HY13-1553-002 US, 2006, www.parker.com

Pawlak AM, Nehl TW, Transient finite element modelling of solenoid actuators: the coupled power electronics, mechanical and magnetic field problem, IEEE Transactions on Magnetics, Vol 24, No. 1, pp. 270-273, January 1988

Payne GS, Rampen WHS, Stein UBP, Ehsan M, Caldwell NJ, Potential of Digital Displacement Hydraulics for Wave Energy Conversion, 6th European Wave and Tidal Energy Conference Glasgow, UK, August 29th - September 2nd 2005

Petersen WA, Hanna TL, Weber JA, Efficiency comparison of axial-piston pumps, Trans. ASME, Vol. 14, pp.72-77, 1971

Petterson M, Andersson J, Krus P, Methods for discrete design optimisation, Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference - DETC2005, Long Beach CA (USA), pp. 295-303, 24-28 September 2005

Pohl J, Sethson M, Krus P, Palmberg J, Modelling and simulation of a fast 2/2 switching valve, Fifth Intl. Conf. on Fluid Power Transmission and Control (ICFP2001), Hangzhou (China), Paper 2-22, 3-5 April 2001

Pohlmann KC, Principles of digital audio (third edition), McGraw-Hill, 1995

Prieto R, Ostergard L, Cobos JA, Uceda J, Axisymmetric modelling of 2D magnetic components, IEEE Applied Power Electronics Conference and Exposition - APEC. vol. 1., Dallas (USA), pp.213-219, March 14-18 1999

Puchalsky C, Megli T, Tiller M, Trask N, Wang Y, Curtis E, Modelica applications for camless engine valvetrain development, 2nd Intl. Modelica Conference, Oberpfaffefhofen (Germany), pp.77-86, March 18-19 2002

Rahman MF, Cheung NC, Lim KW, Position estimation in solenoid actuators, IEEE Transactions on Industry Applications, Vol. 32, No. 3, pp.552-559, May/June 1996

Rameda A, Vilenius J, Huhtala K, Closed loop velocity control of a small teleoperated hydraulic machine, Bath Workshop on Power Transmission and Motion Control PTMC2003, Bath (UK), pp. 105-108, 2003

Rampen W.H.S, Ehsan Md, Almond J.P, Taylor J.R.M and Salter S.H, "Progress on the Development of the Wedding-Cake Digital Hydraulic Pump-Motor", Proceedings of 2nd European Wave Power Conference, Lisbon (Portugal), pp. 289-296, 8-10 Nov. 1995

Rampen W.H.S, Ehsan Md. and Stein U, "A Digital-Hydraulic Power Take-off for Stirling-Cycle Engines", Proceedings of 8th International Stirling Engine Conference and Exhibition, No. ISEC 97083, Ancona, pp.601-612, 27-30 May 1997

Rampen WHS and Salter SH, "The Digital Displacement Hydraulic Piston Pump", Proc. 9th Intnl. Symp. on Fluid Power, BHR Group, Cambridge, STI, pp. 33-46, April 1990

Rampen WHS, Almond JP, Salter SH, "The digital displacement pump/motor operating cycle: experimental results demonstrating the fundamental characteristics", 7th Bath Intnl. Fluid Power Workshop (Innovations in Fluid Power), RSP, Bath, pp. 321-331, 1994

Rampen WHS, The digital displacement hydraulic piston pump (PhD Thesis), University of Edinburgh, December 1992

Rampen, WHS and Salter, SH "Measuring and predicting the frequency response of the digital hydraulic pump", 5th Bath Int'l Fluid Power Workshop (Circuit, Component and Systems Design), Bath, RSP, pp. 155-166, Sept. 1992

Rampen, WHS, Salter SH and Fussey A, "Constant Pressure Control of the Digital Displacement Pump", 4th Bath Intnl. Fluid Power Workshop (Fluid Power Systems and Modelling), Bath (UK), RSP, pp. 45-62, Sept. 1991

Rauch SA, XLXtrFun: Extra functions for Microsoft Excel, Advanced Systems Design and Development, 1999, PA (USA), http://www.xlxtrfun.com/XlXtrFun/XlXtrFun.htm

Reichart M, Murrenhof H, New concepts and design of high response hydraulic valves using piezo-technology, Power Transmission and Motion Control PTMC2006, Bath (UK), pp. 401-414, 2006

Ricco M, De Matthaeis S, Olabi AG, Simulation of the magnetic properties for common rail electro-injector, Journal of Materials Processing Technology, vol. 155-156, pp. 1611-1615, November 2004

Richard CW, Tilley DG, Tomlinson SP, Burrows GR, A second generation package for fluid power systems, Fluid power – Proc. of the 9th Intl. Symp., STI, Oxford (UK), pp. 315-322, 1990

Roters HC, Electromagnetic Devices, John Wiley &Sons, 1945

Salter SH and Rampen WHS, "The Wedding Cake Multi-Eccentric Radial Piston Hydraulic Machine with Direct Computer Control of Displacement", Proc. 10th Intnl. Conf. On Fluid Power, BHR Group, Brugge, MEP, pp. 47-64, April 1993

Salter SH, Rampen WHS, Improved fluid-working machine, European patent EP0494236, 15 July 1992

Salter SH, Rampen WHS, Pump control method and poppet valve therefore, European patent EP0361927, 18 April 1990

Salter SH, Taylor JRM, Caldwell NJ, Power conversion mechanisms for wave energy, Proceedings of the Institution of Mechanical Engineers Part M: Engineering for the Maritime Environment, Vol 216, pp.1-27, 2002

Sangha PS, Rodger D, Design and analysis of voltage fed axisymmetric actuators, IEEE Transactions on Magnetics, Vol 30, No. 5, pp. 3240-3243, September 1994

Sauer Danfoss Inc., Transmission Circuit Recommendations Applications Technical Information (Rev. A July 1997), www.sauer-danfoss.com

Sauer-Danfoss Inc., Datasheet - MCV105V Electronic displacement control, 1999, www.sauer-danfoss.com

Sethson KMR, Vaughan ND, A model of the electromagnetic characteristics of fast switching on/off valves, Sixth Bath International Fluid Power Workshop, Bath (U.K), pp.6-17, 23-24 September 1993

Shengchang Z, Yangzeng X, Guanglin S, Tingxiu Z, Study on extra-high speed digital valve, Fifth Intl. Conf. on Fluid Power Transmission and Control (ICFP2001), Paper 2-35, Hangzhou (China), 3-5 April 2001

Staffa motors sales literature, Kawasaki Motors Corp., Grand Rapids MI (USA), http://www.kawasakipmd.com/pdf/Sales.pdf

Stein UBP, Caldwell NJ, Rampen WHS, Fluid-working machine and operating method, European patent EP1537333, 2005

Stroempl P, Stricklin C, An automotive style hydrostatic transmission control, 1985 SAE International Off-Highway & Powerplant Congress & Exposition, Milwaukee WI (USA), p33-37 (SAE851505), 1985

Szente V, Vad J, Computational and experimental investigation on solenoid valve dynamics, 2001 IEEE Intl. Conf. Advanced Intelligent Mechatronics, Como (Italy), pp. 618-623, 8-12 July 2001

Tilley DG, Tomlinson SP, Practical experience of simulating industrial fluid power systems, 4th Bath Intnl. Fluid Power Workshop (Fluid Power Systems and Modelling), Bath, pp. 315-325, Sept. 1991

Wang S, Miyano T, Hubbard H, Electromagnetic field analysis and dynamic simulation of a two-valve solenoid actuator, IEEE Transactions on Magnetics, Vol 29, No. 2, pp. 1741-1746, March 1993

Wilson WE, Performance criteria for positive displacement pumps and fluid motors, ASME semi-annual meeting, Paper No. 48-SA-14, 30 May – 5 June 1948

Woolman J, Mottram RA, Mechanical and Physical Properties of the British Standard En Steels v.1, Elsevier, 1966

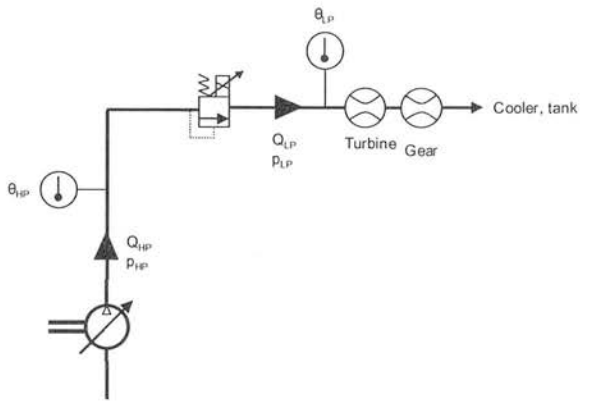
Wu B, Lin C, Filipi Z, Peng H, Assanis D, Optimal power management for a hydraulic hybrid delivery truck, Vehicle System Dynamics, v 42, n. 1-2, pp. 23-40, July-Aug. 2004

Zavarehi MK, Lawrence P, Nonlinear modelling and validation of solenoid-controlled pilot-operated servo-valves, IEE/ASME Transactions on Mechatronics, Vol. 4, No.3, pp.324-334, September 1999

9 Appendices

9.1 Measuring flow at low pressure for efficiency tests

In efficiency testing of hydraulic pumps, it is often preferable to measure the flow downstream of the relief valve. This was the case for the SD1B tests as shown below.



However this requires that the effects of fluid compressibility and thermal expansion be compensated for to obtain an estimate for the actual flow delivered by the pump at high pressure. ANSI/NFPA 3.9.17 gives an equation to compensate for these effects:

$$Q_{HP} = Q_{LP}\left[1 - \left(\frac{p_{HP} - p_{LP}}{k_t}\right) - a(\theta_{LP} - \theta_{HP})\right] \tag{55}$$

where k_t is the bulk modulus, a is the coefficient of thermal expansion, and θ is the fluid temperature. The subscripts LP and HP denote low pressure and high pressure respectively. In the SD1B tests, pump inlet temperature was controlled to 49°C +/- 2°C. Pump outlet temperature was sampled as part of the general data set, but temperature at the flow meter was not measured. The increase of fluid temperature that can be expected from the dissipation of the fluid's pressure energy in the pressure relief valve (Rydberg 2001) can be calculated to a first order approximation:

$$\theta_{LP} - \theta_{HP} = \frac{p_{HP} - p_{LP}}{\rho \cdot c_p} \tag{56}$$

where cp is the fluid heat capacity in J/kgK.

9.2 Special considerations when testing DDP efficiency

Typically efficiency measurements for hydraulic pumps take place at steady states, by averaging the results over some finite time at each condition. The overall efficiency of the pump is the average output power divided by the average input power, where:

Average input power = average (instantaneous torque * instantaneous speed)

Average output power = average (instantaneous flow * instantaneous pressure)

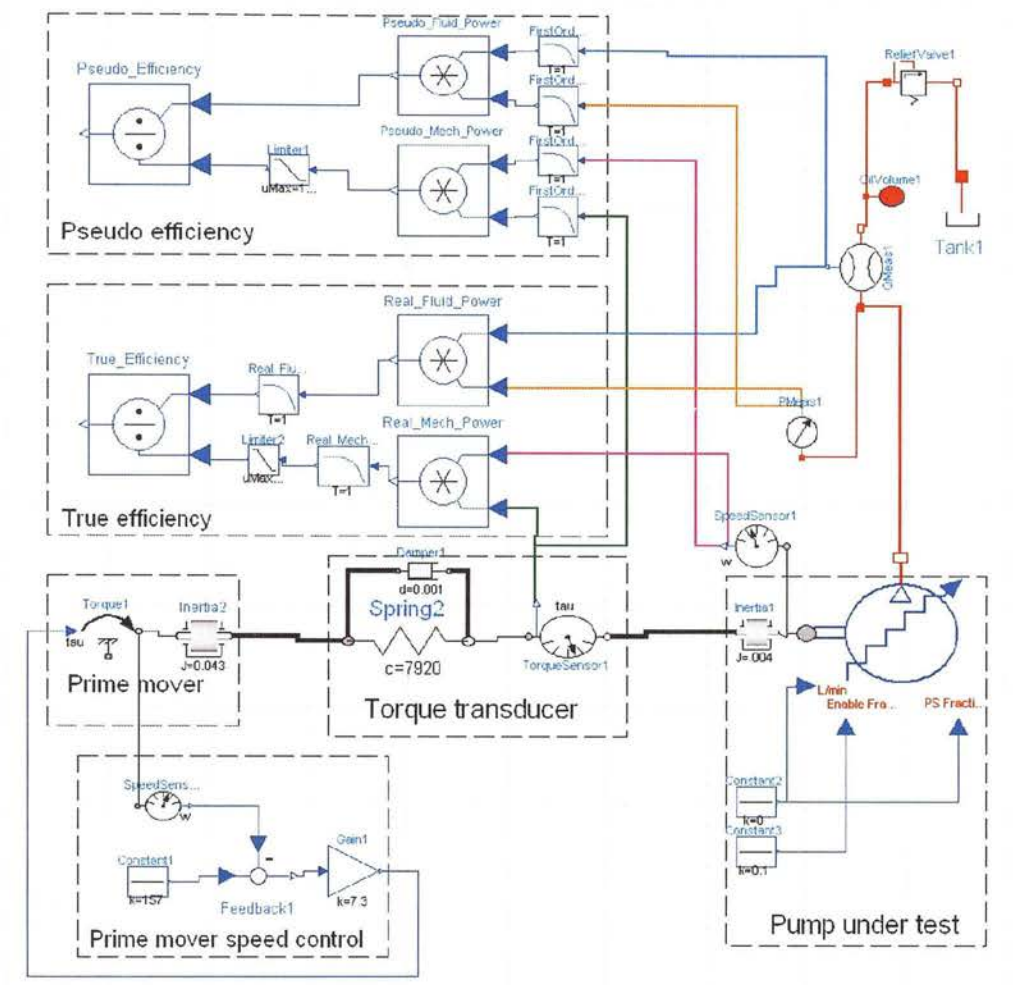
However in such tests it is common practise to filter the signals from all the transducers with low pass filters, with a cutoff frequency of typically around 10Hz. The product of the averaged values from these transducers can be termed the "pseudo average":

"Pseudo average" input power = Average (torque) * Average (speed)

"Pseudo average" output power = Average (flow) * Average (pressure)

The DDP method of displacement control implies that in steady-state conditions of flow and pressure, large instantaneous fluctuations of shaft torque and output flow may occur. When all the measurements relevant to efficiency measurement - pressure, flow, torque and speed - are subject to significant fluctuations, consideration of the detailed measurement of these variables becomes important to arrive at a true measure of the input and output power.

The importance of the distinction between "average" and "pseudo-average" efficiency can be demonstrated with a simple simulation. A model of a DDP in an efficiency testing circuit was created in Dymola. The pump was connected to a hydraulic circuit comprising a compliance and a pressure relief valve.



For this simulation the DDP had 6 cylinders and a total displacement of 12cc/rev.

The pump was driven by an electric induction motor - the parameters for the prime mover of I=0.043 kgm2 and speed stiffness 7.3Nm/rad/s are chosen to model a typical 2 pole 15kW induction motor suitable for testing a 12cc/rev pump (Siemens datasheet 8120065T).

The torque transducer stiffness of 7920Nm/rad stiffness is typical of a commercially available strain gauge model Himmelstein 28002, rated at 113Nm (Himmelstein datasheet 761N).

A range of scenarios was run through the simulation. Motors were considered that were sized appropriately (15kw) and oversized by a factor of two (30kw, I=0.15) and four (60kw, I=0.45). The minimal compliance case (0.20l of oil) considers the case where only a short length of rigid pipe connects the pump to the relief valve, while the middle case (5.0l of oil) models the case with a short length of flexible hose, and the largest compliance is a 1.0l accumulator with a 150 bar gas pre-charge.

The results of this simulation are shown below.

DDP	Motor inertia	Compliant volume, L	Result: Real efficiency	Result: Pseudo efficiency	Error
Perfectly smooth pump	0.043	0.20 oil	1.0000	1.0000	0
6 cyl DDP, 10% enabling	0.043	0.20 oil	1.0000	0.9003	10.07%
6 cyl DDP, 10% enabling	0.043	5.00 oil	1.0000	0.9919	0.81%
6 cyl DDP, 10% enabling	0.150	5.00 oil	1.0000	0.9950	0.50%
6 cyl DDP, 10% enabling	0.450	5.00 oil	1.0000	0.9969	0.31%
6 cyl DDP, 10% enabling	0.450	1.00 gas p/c=150bar	1.0000	0.9996	0.04%
6 cyl DDP, 10% enabling	0.043	1.00 gas p/c=150bar	1.0000	0.9941	0.59%

At the end of 10 seconds of simulation, the calculated efficiency of the DDP from both methods is compared. It is obvious that for the pseudo-efficiency calculation to be acceptably accurate, it is necessary to have both a high hydraulic system compliance and an oversized prime mover.

Unfortunately the shortcomings of commonly-used instrumentation and data acquisition techniques mean that measurement of the truly instantaneous value of some of these variables is problematic. The temptation to simply filter all of the variables separately before multiplying them together to create a "pseudo average" must be resisted unless fluctuations can be controlled to a negligible level.

The torque pulsations may cause significant shaft speed variations – these are obviously worst if the inertia on the pump shaft is low and if the stiffness of the prime mover speed control is low. A.C. induction motors running from a fixed frequency power supply will typically exhibit a 2%-5% speed droop between no-load and full load. This droop can be reduced with the use of a feedback encoder on the motor shaft and a power converter with PID speed control loop. Usually the majority of the inertia in the driveline is located inside the prime mover, so the mechanical stiffness of the shaft and any couplings between the pump and the prime mover must be kept as high as possible and backlash minimised.

Flow pulsations may cause pressure pulsations on the output port, depending on the type of hydraulic load being driven by the pump. A fixed orifice mounted close to the pump outlet is particularly unsuitable because it will produce large pressure fluctuations in response to the flow fluctuations. At first glance a pressure relief valve seems to be the ideal load, but the behaviour of such a valve in transient flow may be non-ideal as a result of the finite mass and damping on the relief valve moving element, and the finite leakage of the valve below its set-point. A good way of keeping output pressure as steady as possible is the combination of a significant compliance (such as a gas accumulator) and a relief valve.

Considering the transducer characteristics:

Torque: Commonly available torque transducers have a signal bandwidth of at least 2kHz, sufficient to capture the half-sinusoidal torque pulsations from a DDP. They do however introduce a significant amount of mechanical compliance in the driveline between prime mover and pump. If the inertia of the driveline on the pump side is high, torque pulses from the DDP will cause "ringing" of this torsional vibration mode.

Pressure: These transducers are usually based on strain gauges. Often they are electrically filtered internally to remove fundamental natural frequency of mass/spring of diaphragm. 2kHz bandwidth or more is easily attainable, however "snubbers" fitted at the input port may create a hydraulic filter with a lower cutoff frequency than the electrical filter.

Speed: Typically this is based on pulses from an optical encoder. There are two methods to process these pulses:

- counting pulses over a period;
- measuring time between edges.

The d.c. tachogenerator offers a better frequency response than either of these methods but may suffer from calibration error. A combination of encoder and tachogenerator may offer the best solution – the encoder being considered the most accurate way of measuring speed by counting pulses during an accurate gate period, the tacho being used to measure the high-frequency variation of the speed within each revolution.

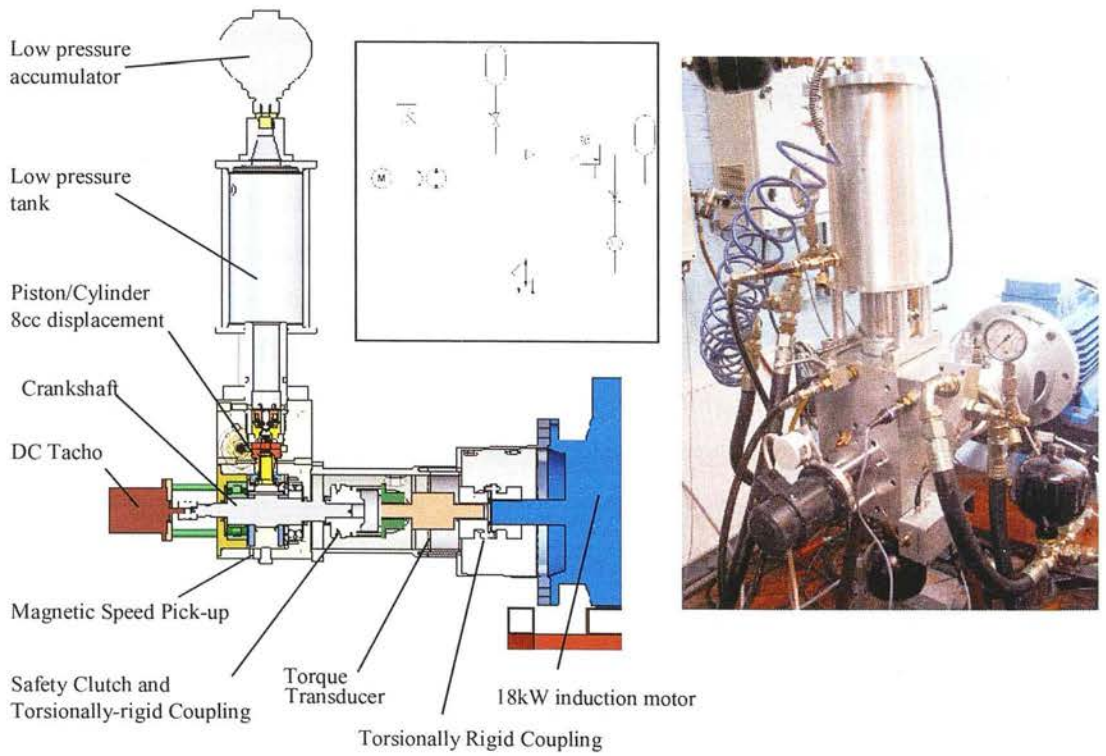
Flow: The displacement resolution of positive-displacement flow sensors is usually so low that there is no chance of resolving individual flow pulses. Therefore the objective of flow measurement can only be to ensure that the sum of pulses counted from the flow sensor equates to the volume of fluid which passes through it.

One option is to process the pulses through a frequency-to-voltage converter, and acquire the analogue signal after an anti-aliasing filter. However it is the experience of the author that with pulsatile flow, instantaneous flow may reverse through the meter. If only one edge

sensor is used, this can cause spurious edges to be counted as the gear "jitters" forwards and backwards. If two edge detecting sensors are fitted out of phase with the gear teeth, pulses can first be quadrature decoded before being counted. However it is the experience of the author that data acquisition electronics adapted to properly process the quadrature signal are not commonly fitted to general purpose efficiency testing rigs. A more generally applicable method is to measure flow downstream of the pressure relief valve, and create a smoothing filter around the flowmeter with an orifice and an accumulator. The actual flow produced by the pump is lower than that recorded with this method because of bulk modulus and thermal expansion.

At the early stage of developing a DDP the most logical way to work is with a single cylinder pump; this minimises the number of duplicate prototype parts that need to be made to test each design iteration. Unfortunately these measurement issues are worst of all in a single cylinder test rig. If the prime mover is sized for the steady state power consumption of the single cylinder then it will have a low inertia, exaggerating the shaft speed variation. Even at full displacement the flow output is pulsatile, being a half-wave sinusoid. This exaggerates the pressure variation.

As an example, consider the single cylinder DDP test rig and results from it shown overleaf[4].



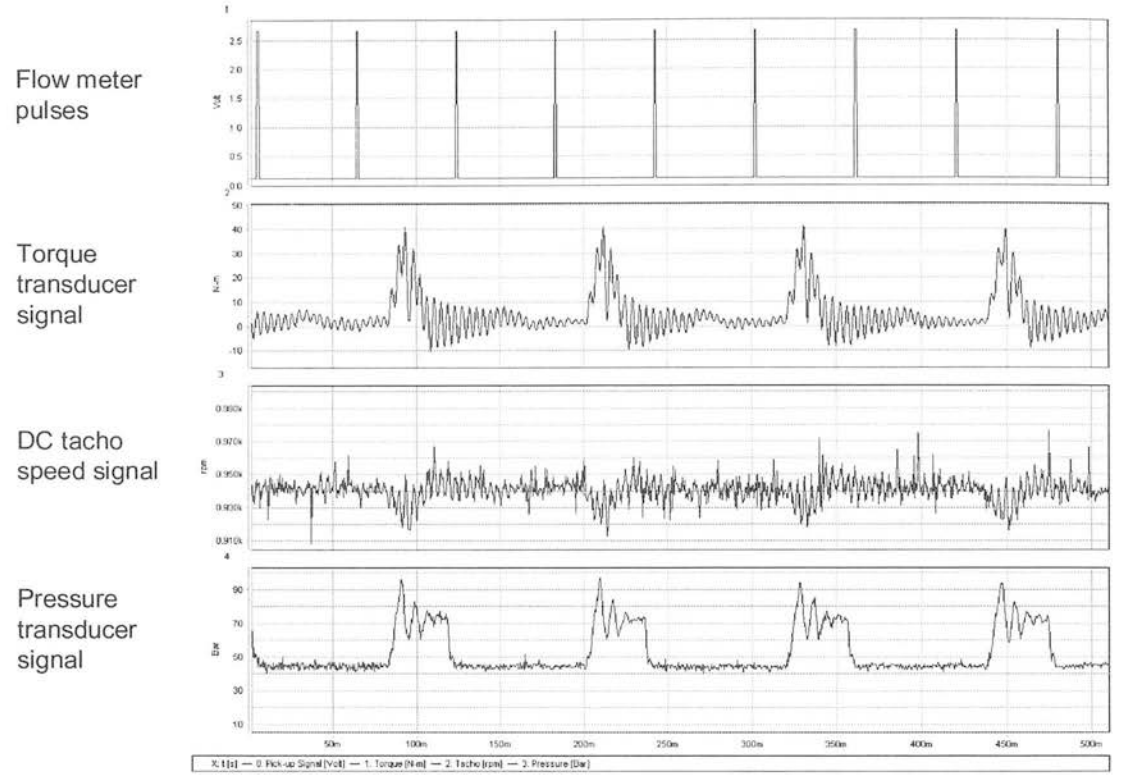
[4] This rig was designed and built by Mr. Craig Lowe under the supervision of the author

The machine under test was a single cylinder radial piston DDP of 8cc/rev. The driveline had significant inertia on the pump side of the torque transducer due to the fitting of a torque-limiting safety clutch; this was needed to protect the torque transducer in case the mechanism jammed. In conjunction with the significant compliance of the torque transducer, this created a torsional resonance which can be clearly seen as ringing in the torque transducer signal at approximately 240Hz. Moving the safety clutch to the motor side of the torque transducer is obviously to be recommended, as is choosing the stiffest possible torque transducer – with the trade-off that this will typically imply a reduced sensitivity. Speed was measured by a tachogenerator.

The schematic shows the arrangement of needle valve and low-pressure accumulator (pre-charge = 5 bar) used to smooth the flow through the flow transducer. At each flow-rate the needle valve was adjusted so that the maximum smoothing effect of the accumulator and orifice was achieved. Not shown in the schematic is a second relief valve around the needle valve set at 20 bar; this is needed for safety in case the needle valve is closed completely. The main relief valve at the pump outlet must be atmospherically vented to prevent the downstream pressure drop of the smoothing circuit from adding to its setting.

The value of the smoothing circuit was established in use; with the needle valve fully open, pulses from the flow transducer had false edges as the gear teeth momentarily changed direction in response to the pulsatile flow. As this transducer had only a single edge detecting sensor, it was impossible to filter out these false edges and the result was a large over-estimate of the actual flow through the transducer. Closing the smoothing needle valve cured this problem completely by ensuring that the flow through the flow transducer was steady, even though the flow from the pump was pulsatile. This underlines the importance of close inspection of the raw pulse signal coming from the flow transducer before edges are counted by digital counting or a frequency to voltage converter.

The raw transducer results from a single-cylinder DDP test are shown below:



The following observations can be made:

- The torque is strongly pulsatile, and "ringing" can be clearly seen in the driveline caused by the torsional spring/mass system of torque transducer compliance and the inertia of the safety clutch and coupling;
- The smoothing circuit has ensured that the flow transducer pulse frequency is steady despite the pulsatile flow from the DDP.
- There is measurable shaft speed variation when the torque pulses are applied
- The pressure trace shows insufficient hydraulic compliance- the pressure is much higher during the pumping stroke than during the intake stroke.

From these observations the following general guidelines for efficiency testing of DDPs can be made:

- With commonly available instrumentation, it is impossible to measure the true efficiency of a DDP due to rapid fluctuations of flow, speed, pressure and torque.
- Therefore, the objective must be to minimise the variation of pressure and speed as much as is possible such that the "pseudo-average" efficiency measured is as accurate as possible.
- To minimise shaft speed variation, the prime mover inertia should be as large as possible

- To minimise torsional vibration, the driveline stiffness (including torque transducer) should be as high as possible, while the inertia on the pump side of the transducer should be minimised.
- To eliminate the possibility of false counting of the flow transducer pulses, quadrature decoding should be used. Either a special digital electronic hardware counter can be used, or if this is not available then the two quadrature phases can be sampled separately along with the analogue signals, with quadrature decoding and pulse counting happening at the post-processing stage. Measurement downstream of the pressure relief valve with a smoothing circuit helps keep steady flow through the transducer even though pump flow is pulsatile, but correction for compressibility and temperature must then be applied.
- The hydraulic compliance should be as high as possible, and situated as close as possible to DDP outlet to minimise pressure pulsation.

9.3 Applying the DDP loss model to backward-facing simulation studies

The forward and backward-facing formulations for vehicle propel simulations have different strengths and weaknesses, and different requirements for the form of the component loss models. This note outlines how the forward-facing DDP loss model presented here may be converted to work in a backward-facing simulation.

Forward facing formulation

The inputs and outputs of a forward-facing pump model are shown below:

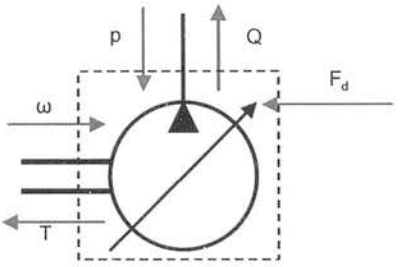


Figure 9-1: Inputs and outputs of a forward-facing pump model

In a forward-facing vehicle simulation, the vehicle is not constrained to follow the input drive cycle exactly. Instead, a virtual driver is given speed demand and must manipulate the vehicle controls such that the actual vehicle speed matches the demand within a defined error range. The engine speed is the result of a dynamic model of the engine inertia, the torque added by the engine (typically under command of a speed governor) and the torque extracted by the pump. The pump displacement is controlled by the virtual driver or by a virtual system controller reacting to driver demand; the dynamics of the pump control mechanism can be incorporated. The pump delivers flow into a compliance, the pressure resulting from the integral of flow in and flow out being fed back to the pump. The motor produces torque as a function of the pressure in the compliance, this torque being applied to a 1D dynamic vehicle model where the vehicle mass is accelerated by the difference between the tractive effort produced at the wheels by the motor torque and the drag force. The speed of the vehicle is determined by the output of the integrator, and is fed back to the motor. The motor uses this speed signal to determine the flow taken from the pressure-buildup.

There are typically three main numerical integration blocks: vehicle mass, hydraulic stiffness and engine inertia (Lennevi and Palmberg 1995). In addition there are feedback control loops formed by the virtual driver trying to control the vehicle to match the demanded speed, the

engine governor trying to control engine speed, and any nested electro-hydraulic control loops in the hydrostatic machine swash angle control system.

In general the loss models for axial piston machines are forward-facing; one of the main motivations for these models is to be able to simulate HST control functions (Huhtala et al. 1995).

Forward-facing vehicle model: strengths (+) and weaknesses (-)

+ Can model transient behaviour and drivability

+ Allows model-based design of the control system

- Very slow simulation due to the stiffness of the hydraulic compliance; parametric optimisation of fuel consumption of whole drive-cycles is difficult as a result

- A "virtual driver" is required which attempts to satisfy the demands of the duty cycle

- A virtual control system must be designed and optimised before drive-cycle fuel consumption can be calculated.

- A variable-time step solver is required to cope with numerical stiffness and discontinuities in a typical HST

Backward-facing formulation

The inputs an outputs of a backward-facing pump model are shown below:

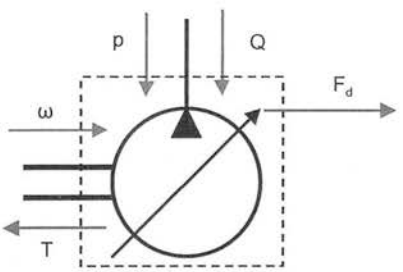


Figure 9-2: Inputs and outputs of a backward-facing pump model

The backward-facing form of simulation is well-established in the automobile drive-cycle modelling community. The vehicle is constrained to follow an input time series of speed. At each time step, the acceleration required of the vehicle gives the force required to accelerate the mass, while polynomial coastdown coefficients model drag from aerodynamics, driveline losses, and tyre rolling resistance. The sum of the inertial and drag forces determines the torque output required of the motor, while its speed is fixed by the vehicle speed. The loss

model must calculate the input flow and pressure required for the motor to create this output flow and pressure. The pump is constrained to deliver the demanded flow and pressure at the current engine speed; the loss model calculates the torque imposed on the shaft and the displacement fraction required to produce the demanded flow. The prime mover is assumed to be rotating at the speed demanded by the vehicle controller, often chosen with reference to a look-up table or map to give the optimum fuel consumption for the instantaneous power demand. The torque exerted by the pump on the engine shaft is determined by the pump loss model.

The model is "backward facing" because the vehicle is constrained at each time step to be at a demanded speed; the energy flows are calculated backwards up the driveline towards the engine. Because there are no feedback loops, there is no integrator, so the time steps can be very long relative to forward-facing models: in the author's experience there is no benefit to time steps smaller than 0.1 second, allowing a typical 20 minutes duty cycle to be modelled in a few seconds of real time. This allows the very rapid parametric optimisation of overall fuel consumption as a function of machine displacement or other parameters.

Backward-facing vehicle model: strengths (+) and weaknesses (-)

+ very fast solution; allows rapid parametric optimisation

+ simple compared to a forward-facing model

+ no need for dynamic control system or virtual driver

- no ability to model transient behaviour or drivability

- does not allow control system prototyping

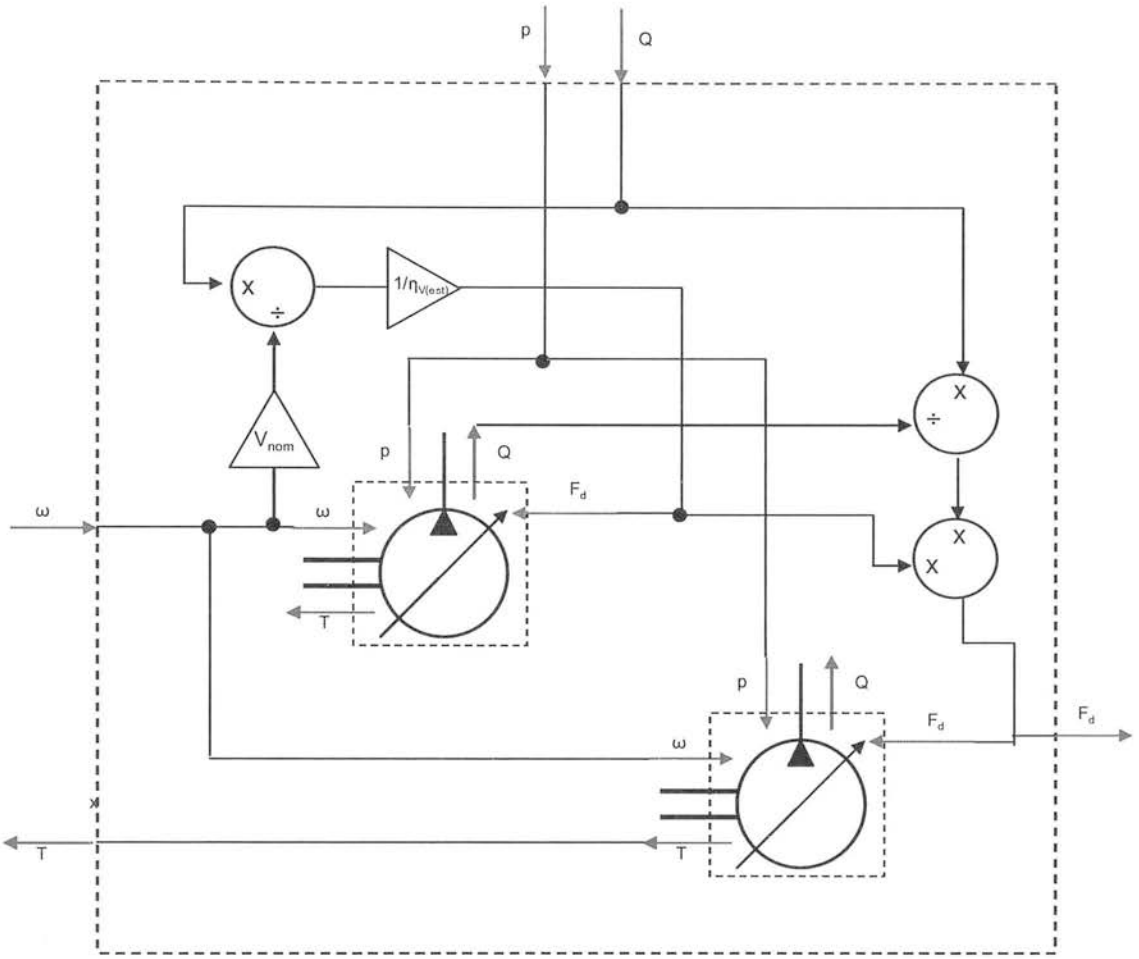
- does not prove control system will be stable

- model fails if vehicle cannot attain commanded speed due to power limits

Adapting a forward-facing component model to work in a backward-facing model

The DDP loss model presented here is forward-facing. It may be useful to adapt such a model to backward-facing simulations. Here the challenge is to determine the displacement fraction at which the DDP operates to satisfy the output flow demand; in the forward-facing model, pump output flow is an output rather than an input.

This objective may be achieved by using a two-step iteration, shown diagrammatically below.



In the first step, the displacement fraction is estimated by dividing the output flow demand by the product of pump maximum displacement and speed. This estimate is fed to the forward-facing model which calculates the actual output flow at this displacement fraction. The ratio between this flow and the demand flow is used to scale the displacement fraction estimate, and this scaled estimate is passed again to the forward-facing model. Because of the linearity in the model between displacement fraction and output flow, the resulting output flow of the second forward-facing model matches exactly the flow demand.

This linearity of flow to displacement fraction is unique to the DDP; for simulation of axial-piston machines, many more iterations may be required to achieve the output flow with acceptable accuracy.

Note that in the backward-facing formulation, it may be that the pump displacement required to satisfy the flow demand exceeds unity; checks should be added to the simulation code to catch this error condition, which indicates that the pump is undersized or the input speed is too low.

9.4 Development of the LPV poppet attachment

During development of the LPV for the propel DDPM, three methods a were investigated for the attachment between the armature and the poppet.

The first attempt at the attachment between poppet and armature had previously failed on the DDP LPVs due to impact fatigue of the narrow ledges which located the armature axially. The armature, being steel, has considerable kinetic energy when the poppet hits the seat. During the impact, this energy has to be absorbed by the axial elastic strain of the poppet between seat and armature, and transferred across the locating faces to the armature. The first design, shown below, used 4 narrow ledges for location, and was retained by staking the armature to the poppet – the armature having narrow deformable lugs formed with narrow saw cuts.

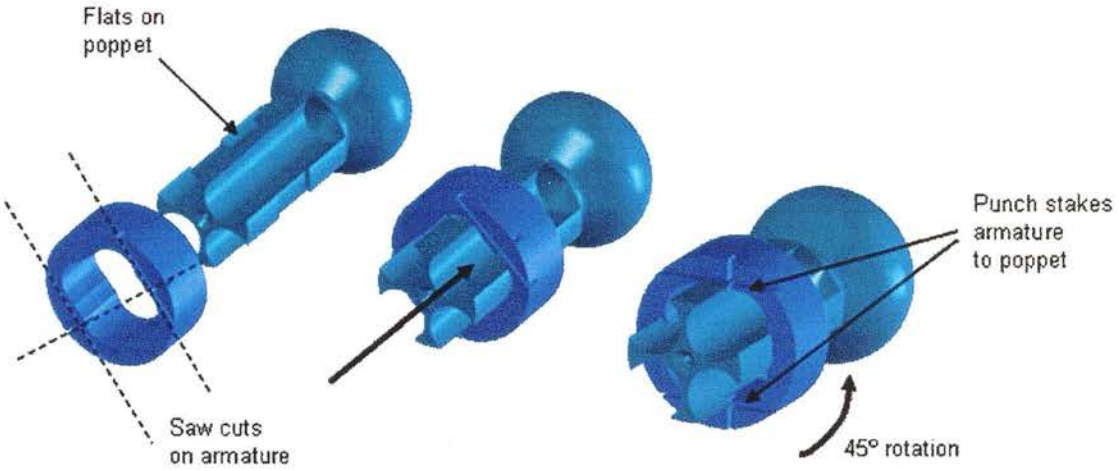


Figure 9-3: Early attachment means for LPV

Inspection of failed parts indicated that the staking operation had damaged the poppet material, weakening it to the extent that the attachment would get loose after a few hours' operation, causing axial play in the attachment.

An improved second arrangement was used for LPV for the DDPM-P. Through slots were milled part way along the length of the poppet, effectively creating 4 elastic beams. The armature and poppet were designed with a much increased axial contact area to reduce the impact stress, while the deflection of the beams was used to provide a "snap fit". A controlled amount of interference (0.10mm radial) generated a positive locking action that prevented the armature from rotating:

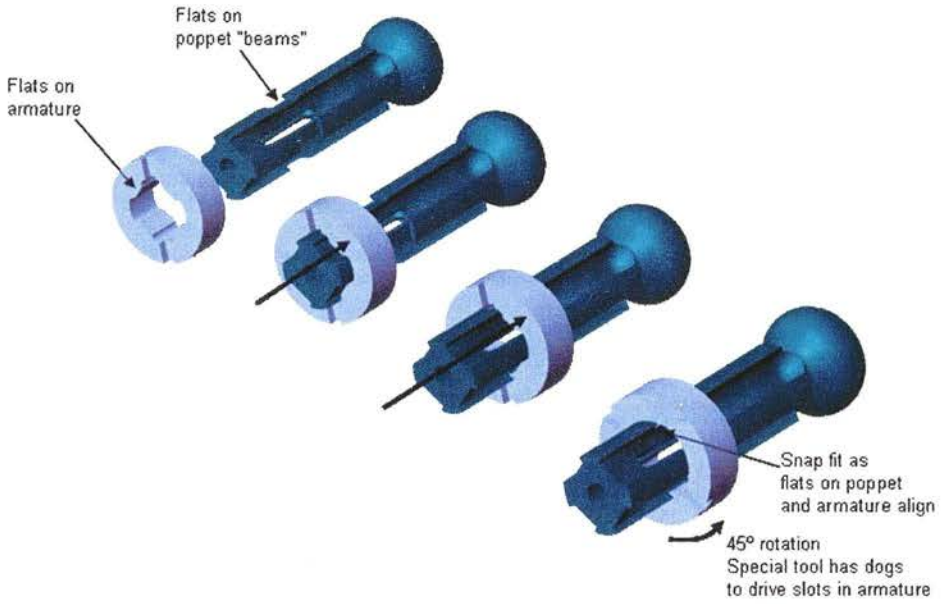


Figure 9-4: "Snap fit" concept of armature attachment for the LPV

This "snap fit" attachment was successful in eliminating the axial play caused by fretting wear of the "staked" design. However it was difficult to achieve the necessary machining tolerances to ensure that the interference of the fit was large enough to withstand the forces in operation, but not so large as to cause deformation of the cylindrical guiding surface, which caused jamming of the poppet in the guide bore. This experience led to the development of the "glued tube" attachment method which is detailed in section 3.3.4.

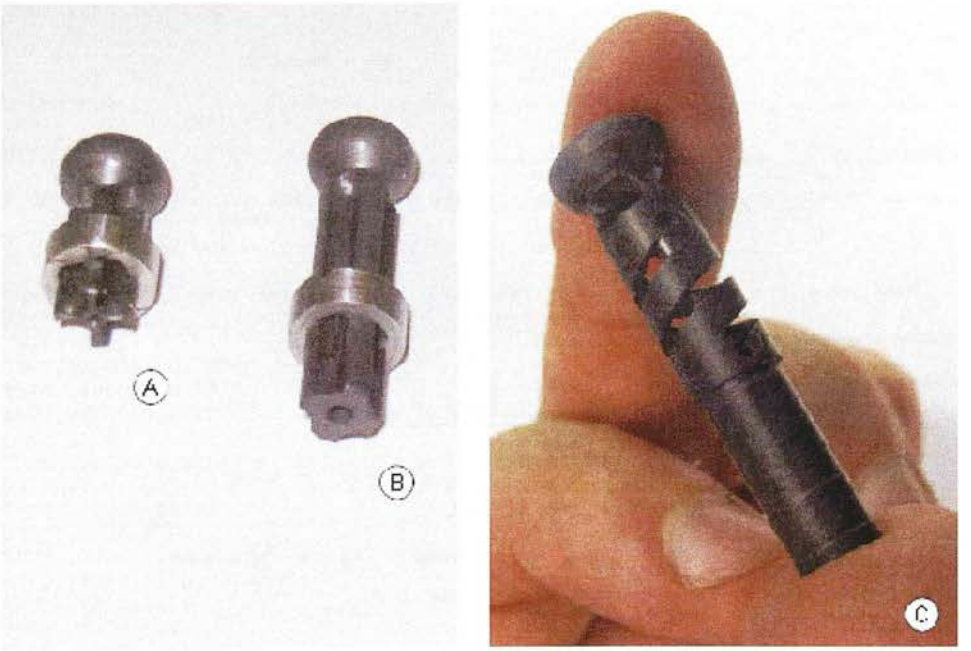


Figure 9-5: Evolution of LPV poppet design; (A) Staked; (B) Snap fit; (C) Glued tube

9.5 Mathcad calculations for capillary impedance design

12mm pad bore, with the 19mm square pads and an impedence comprising 1.19mm wire in a 1.30mm hole, 20 mm long. These are clearly underclamped without the impedence. The large land flowpath gives lower leakage for a given clearance than a thin land bearing. Pressure distribution across the land is not clearly known, but BHRA equations predict 15% underclamp with these dimensions.

PadBore=0.012, landthick 0.0035.

Leakage flow rates at 200 bar desired about 0.2 cc per second. Pressure drop at this flowrate to be 20 bar.

Capillary restrictor

Oil Kinematic Viscosity

$$OilKViscCST := 29$$

$$OilKVisc := OilKViscCST \cdot 0.000001$$

$$OilKVisc = 2.9 \times 10^{-5}$$

OilDensity

$$OilDensity := 900$$

Oil Dynamic Viscosity

$$OilDVisc := OilKVisc \cdot OilDensity$$

Capillary Length

$$L := 0.020$$

Hole Diameter

$$Dh := 0.00131$$

Set to 1.31mm for agreement with results

Wire Diameter

$$Dw := 0.00119$$

Flow area

$$A := \left(\frac{Dh^2}{4} - \frac{Dw^2}{4} \right) \cdot \pi \qquad A = 2.356 \times 10^{-7}$$

$$Perimeter := (Dh + Dw) \cdot \pi \qquad Perimeter = 7.854 \times 10^{-3}$$

Hydraulic mean diameter

$$Dhm := \frac{4 \cdot A}{Perimeter} \qquad Dh = 1.31 \times 10^{-3}$$

Hydraulic Area

$$Ah := \frac{Dhm^2 \cdot \pi}{4} \qquad Ah = 1.131 \times 10^{-8}$$

Target leakage of one cylinder volume in ten seconds at 200 bar

Flow Rate

$$Flow := 0.02 \cdot 10^{-6}, 0.04 \cdot 10^{-6} .. 1.0 \cdot 10^{-6}$$

Average Velocity

$$v(Flow) := \frac{Flow}{A} \qquad v = function$$

$$v\left(0.1 \cdot 10^{-6}\right) = 0.424$$

Reynold number

$$Re(Flow) := \frac{v(Flow) \cdot Dhm}{OilKVisc} \qquad Re = function$$

$$Re\left(0.1 \cdot 10^{-6}\right) = 1.756$$

Friction factor, laminar

$$Fric(Flow) := \frac{64}{Re(Flow)}$$

Bernoulli pressure drop

$$Pb(Flow) := \frac{OilDensity \cdot v(Flow)^2}{2} \qquad Pb = function$$

$$PbBar(Flow) := \frac{Pb(Flow)}{100000} \qquad PbBar\left(0.1 \cdot 10^{-6}\right) = 8.106 \times 10^{-4}$$

Note this is a capillary, not an orifice, so Bernoulli pressure drop should be negligible

Pressure drop, laminar flow, from Douglas

$$PLam(Flow) := Fric(Flow) \cdot \frac{L \cdot OilDensity \cdot v(Flow)^2}{2\,Dhm}$$

$$PLam = function$$

$$PLamBar(Flow) := \frac{PLam(Flow)}{100000}$$

$$PLamBar\left(0.2 \cdot 10^{-6}\right) = 9.846$$

$$PTotalBar(Flow) := PLamBar(Flow) + PbBar(Flow)$$

Impedence constant, cc/s per bar

$$ImpConst := \frac{0.2}{PLamBar\left(0.2 \cdot 10^{-6}\right)} \qquad ImpConst = 20.312 \times 10^{-3}$$

9.6 ANSYS input deck for the HPV analysis

```
! DDPM high pressure valve magnetic analysis
! (c) 1999-2000 Niall Caldwell, Artemis Intelligent Power Ltd.

*set,axgap,0.00001     ! set gap between lower and moving pole
*set,rgap,0.00015     ! radial clearance
*set,magl,0.0025       ! Axial length of perm magnet
*set,magr,0.001      ! Radius of erm magnet
*set,thk,0.0015        ! set moving pole web thickness
*set,coilir,0.0065     ! Coil internal radius
*set,poleir,0.002     ! Moving pole internal radius
*set,poleer,0.0065     ! Moving pole external radius
!*SET,current,0.2    ! Current in HP coil
*set,aturns,432*current ! Calc ampere-turns
*set,ringir,0.0095   ! Inside radius of radial flux ring
*set,thinmag,0.003     ! Thickness of thin magnetic poleface
*set,taperw,0.0005     ! Width of tapered part of moving pole face
*set,taperh,0.001   ! Height of tapered part of moving pole face

/PREP7
/title, HPV MAGNETIC LATCH

ET,1,13,,,1            ! Use Plane13 axisymmetric elements
EMUNIT,MKS,            ! Units to be mks.

!Define material properties

MP,MURX,1,1            ! Air

TB,BH,2,,,              ! data for B-H curve for steel EN1a, material 2
TBPT,,800,0.48
TBPT,,1600,1.04
TBPT,,3200,1.42
TBPT,,4000,1.50
TBPT,,4800,1.56
TBPT,,6400,1.67
TBPT,,8000,1.72
TBPT,,12000,1.81
TBPT,,16000,1.88
TBPT,,20000,1.935
TBPT,,300000,2.285



MP,MURX,3,1            ! EN58 assumed to be air (valid as long as not cold worked)

MP,MURX,4,1            ! Coil

! data for B-H curve for sintered samarium cobalt, material 5

! data for B-H curve for sintered NeFeBo, material 5

hc=800000  ! coercive force.
MP,MGYY,5,HC       !FLUX DIRECTION - UPWARDS
TB,BH,5,,,
TBPT,,200000,0.295
TBPT,,400000,0.59
TBPT,,800000,1.18

! Define pole material serperately for subsequenct easy picking
TB,BH,6,,,              ! data for B-H curve for steel EN1a, material 6
TBPT,,800,0.48
TBPT,,1600,1.04
TBPT,,3200,1.42
TBPT,,4000,1.50
TBPT,,4800,1.56
TBPT,,6400,1.67
TBPT,,8000,1.72
TBPT,,12000,1.81
TBPT,,16000,1.88
TBPT,,20000,1.935
TBPT,,300000,2.285
```

```
! Material data over

K,1, 0.0, 0.0, 0.0
K,2, 0.008, 0, 0
K,3, 0.008, 0.002, 0
K,4, 0.010 , 0.002 , 0
K,5, ringir , 0.0103-taperh/2-thk-0.0025 , 0
K,6, 0.010 , 0.0103-taperh/2-thk-0.0025 , 0
K,7, 0.014 , 0.0075-axgap-thk-0.0005 , 0
K,8, ringir , 0.0103+(taperh/2) , 0
K,9, 0.01 , 0.0103+(taperh/2) , 0
K,10, 0.011 , 0.015 , 0
K,11, 0.011 , 0.026+magl , 0
K,12, 0.014 , 0.029+magl , 0
K,13, 0.00 , 0.029+magl , 0
K,14, 0.00 , 0.026+magl , 0
K,15, magr , 0.026+magl , 0
K,16, coilir, 0.026+magl, 0.0
K,17, coilir, 0.026, 0.0
K,18, magr, 0.026, 0.0
K,19, 0.0, 0.026, 0.0
K,20, 0.0, 0.025, 0.0
K,21, poleir, 0.0233, 0.0
K,22, poleir, 0.0103+THINMAG, 0.0
K,23, poleer, 0.0103+THINMAG, 0.0
K,24, poleir, 0.0103-axgap-(taperh/2), 0.0
K,25, 0., 0., 0.0                ! null point- made in error
K,26, poleer, 0.0103-axgap-(taperh/2), 0.0
K,27, poleer, 0.0103-axgap-(taperh/2)-0.0001, 0.0
K,28, ringir-rgap, 0.0103-axgap-(taperh/2)-0.0001, 0.0
K,29, ringir, 0.0103-axgap-(taperh/2)-0.0001, 0.0
K,30, ringir, 0.0103-axgap-(taperh/2)-0.0001-thk, 0.0
K,31, ringir-rgap, 0.0103-axgap-(taperh/2)-0.0001-thk, 0.0
K,32, ringir-rgap-thk/2, 0.0103-axgap-(taperh/2)-0.0001-thk, 0.0
K,33, poleer, 0.0103-axgap-(taperh/2)-0.0001-thk, 0.0
K,34, poleer, 0.0103-axgap-(taperh/2)-0.0001-thk-0.0001, 0.0
K,35, poleir, 0.0103-axgap-(taperh/2)-0.0001-thk-0.0001, 0.0
K,36, 0.0, -0.003, 0.0
K,37, 0.014, -0.003, 0.0
K,38, coilir, 0.015, 0.0
K,39, poleir, 0.0103+(taperh/2), 0.0
K,40, poleer, 0.0103+(taperh/2), 0.0
K,41, poleir+((poleer-poleir-taperw)/2), 0.0103-axgap, 0.0
K,42, poleer-((poleer-poleir-taperw)/2), 0.0103-axgap, 0.0
k,43, poleir+((poleer-poleir-taperw)/2), 0.0103, 0.0
k,44, poleer-((poleer-poleir-taperw)/2), 0.0103, 0.0

L,1,2   ! Line 1
L,2,3
L,3,4
L,3,5
L,4,6
L,5,6
L,6,7
L,6,9
L,8,9
L,8,29 ! LINE 10
L,29,30
L,30,5
L,7,12
L,12,13
L,13,14
L,14,15
L,15,16
L,16,11
L,16,17
L,15,18  ! LINE 20
L,18,19
L,19,14
L,17,18
L,17,38
L,38,10
L,10,11
L,9,10
L,19,20
L,20,21
L,21,22  ! LINE 30
```

```
L,22,43
L,24,26
L,23,38
L,22,24
L,23,26
L,24,35
L,26,27
L,27,28
L,28,29
L,30,31 ! LINE 40
L,31,32
L,32,33
L,27,33
L,33,34
L,34,35
L,36,37
L,37,7
L,1,36
L,1,20
L,22,39 ! LINE 50
L,23,40
L,24,41
L,42,26
L,41,42
L,44,40
L,43,44 ! LINE 56
l,39,43

l,43,41 ! 58
l,44,42 ! 59

A, 20,21,22,39,24,35,34,33,32,31,30,5,3,2,1      ! AREA 1
A, 1,2,3,4,6,9,10,11,16,15,14,13,12,7,37,36
A,14,15,18,19
A,15,16,17,18
A,16,11,10,38,17 ! AREA5
A,19,18,17,38,23,22,21,20
A,38,10,9,8,23
A,8,9,6,5,30,29
A,5,6,4,3
A,39,43,44,40,26,42,41,24 ! AREA 10
A,24,41,42,26,27,33,34,35
A,27,28,31,32,33
A,28,29,30,31
A,26,40,23,8,29,28,27
A,22,23,40,44,43,39 ! AREA 15

ASEL,NONE          ! AREAS TO BE AIR
FLST,5,5,5,NOOR,5
FITEM,5,1
FITEM,5,9
FITEM,5,13
FITEM,5,14
FITEM,5,10
ASEL,S, , ,P51X
AATT,1

ASEL,NONE          ! AREAS TO BE STEEL
FLST,5,6,5,NOOR,6
FITEM,5,2
FITEM,5,6
FITEM,5,8
FITEM,5,4
FITEM,5,3
FITEM,5,15
ASEL,S, , ,P51X
AATT,2

ASEL,NONE          ! AREA TO BE EN58
ASEL,S,,,7
AATT,3

ASEL,NONE          ! AREA TO BE COIL
ASEL,S,,,5
AATT,4

!ASEL,NONE          ! AREA TO BE MAGNET
```

```
!ASEL,S,,,15
!AATT,5

ASEL,NONE          ! AREAS TO BE POLE
FLST,5,2,5,NOOR,2
FITEM,5,11
FITEM,5,12
ASEL,S, , ,P51X
AATT,6



FLST,2,9,4         ! PARALLEL FLUX BOUNDARY CONDITION
FITEM,2,14
FITEM,2,13
FITEM,2,47
FITEM,2,46
FITEM,2,48
FITEM,2,49
FITEM,2,28
FITEM,2,22
FITEM,2,15
DL,P51X, ,ASYM

LESIZE,34,,,5,1    ! 5 ELEMENTS SPACED ALONG SPECIFIED LINES (AIR GAPS)
LESIZE,35,,,5,1
LESIZE,39,,,5,1
LESIZE,40,,,5,1

LESIZE,56,,,20,1
LESIZE,54,,,20,1
LESIZE,58,,,5,1
LESIZE,59,,,5,1

smrtsize,1
ASEL, all
AMESH ,ALL         ! AUTO MESH ALL

ASEL,NONE          ! AREAS CONTAINING ELEMENTS TO BE PART OF POLE COMPONENT
ESEL,S,MAT,,6
CM,POLE,ELEM
FMAGBC,'POLE'

!The following procedure calcuates the coil area
!to calculate current density
kdist,16,11
*set,coilw,_kdist
kdist,11,10
*set,coilh,_kdist
*set,curdense,(aturns/(coilw*coilh))
ESEL,S,MAT,,4,4                ! SELECT COIL
BFE,ALL,JS,1, , ,curdense, ,   ! APPLY EXCITATION

/SOLU
MAGSOLV,0,3,0.001, ,25,

/POST1
*SET,LAST      ! Retrieve latest solution set
*SET,_IDBUG,1 ! Allow retrieval of data from fmagsum.mac (hack)
FMAGSUM
*SET,POLEFORC,(_FOR(1,2,2))  ! force on pole from fmagsum
srcs,1
/focus,1,.65E-02,0.103E-01
/dist,1,0.005
/EDGE,1,1,45  ! Switch on element outlines
/GLINE,1,0
/SEG,MULTI     ! Store subsequent plots for animation
PLNSOL,B,SUM,0, ! B nodal plot
/SEG,OFF       ! Stop storing segments
/PREP7
```

9.7 Wheel motor embedded software

```
/* Wheel motor software
(c) 2001 -2002 Niall Caldwell, Artemis Intelligent Power Ltd.

Updates:
selective enabling of serial diagnostics
pot comtrolled valve timing with serial feedback
revised variable allocation to free space
extra logic for total of 8 valve events during FALSE rev
speed sensing
speed related timing - now re-enabled with better timing constants
Bi-directional working
Dump sequence working from pedal pot, optimised delays
Automatic retry of FALSE rev
Active pumping
Ruthless cull of pointless variables
Status checking now hangs off timer 1 overflow interupt to cure 'while loop jump
out'problem.
Sorted latent problem in set_int with offset overflow- better starting
Experimental idle condition [disabled LP OFF if motor_enable==FALSE] - NOW WORKING!
Now has sensing for actual_dir and restart_motoring disabled above 300rpm,
  avoiding jerky reverse motor stops
Trial of half displacement, based on wmotor18.c - goes to half displacement when
     pot is turned fully forward or reverse.
wmotor24 has first attempt at full variable displacement with speed loop - forward
only
wmotor25 as 24 but with speed loop disabled- direct duty cycle control from pot
Now has verified variable displacement
Added hysteresis to LS and HS cross sector timing
wmotoT25 test version of wmotor25 with disabled speed related timing
wmotor27 is torque control in integral steps- full, then half, then third displacment
as
       pot is turned
gmot1 is based on wmotor27, modified for repackaged electronics in initial buggy
trials.
gmot2 adds multiselctor control for pot inputs, replacing check_pots
gmot3 has first try at software buffer for serial port
gmot4 has fully working software buffer
gmot5 integrates speed control loop and pressure command from wmotor 26
gmot6 has progressive speed limit, controlled by pot_bin(9)

gmo6T2 try to cure lpv pulse overlap due to fire_poppet problems
  Inhibits duplicate 'on' pulses when valves already on, using status arrays
 gmot7 cures this problem by shifting check_pots() to flag triggered at BDC rather
than async
            also optional pulse saving enabled at compile, see main()
            resets all pop_no counters at BDC to avoid out of sync problems
            lower rate check_pots() and fewer buff_transmit() calls
            valve event interrupts now absolute from bdc rather than relative from
last one
gmot8 first trial to integrate all 3 pedals including speed demand
             now uses torque / pressure algorithm for speed control
gmot9 now has control of speed loop gains from potbins 5,6,7
            now  use movement to produce regular speed updates under 300 rpm.
gmot11 recreation of gmot10, lost in laptop crash
     Disable direction switch at speed
     Speed up false rev by releasing dump valve halfway through at M=5, dump valve
       delay = 40000, false_rev_delay=10000, disabled extra extend dump delay.
     Implement extra logic at end of false rev to ensure each cylinder has either LPV
or HPV active,
       enabled by full_start
gmot12 has progressive starting pressure, controlled by pedals
       cured crash on reverse caused by BDC during false rev
gmot13 try to double HPV firings during false rev to enhance starting

gmot14 automatic self test procedure set by pot_bin[10] >900 or < 100
  Relief valve on overrun
  Full_start inhibited if potbin[10] < 200
  Restarts inhibited by brake pedal > 600, with pump_pedal allowing extra false revs
on demand

gmot15
```

```
     moved time/position switch-over to last fire_poppet event rather than
overflow_count
     resample ad(3) at last fire_poppet event, print out movement
     print out position error at first BDC after false rev- in form Txxxx
     (defined first_bdc flag to detect this condition)
     taken out doubled HP pulse on false_rev
     compacted code to fit into 1FFF memory
     fixed lp_status problem due to do_valve_event

gmot16

     disabled encoder pulse increment if rolling back by sampling ad(3)

      alter timer1 overflow 'heartbeat' to 10ms from 65ms
        Establish overflow counter, set divisors to get back to the same
           update rates as before
             Changed maxcount to 10000
             Increased overflow_count limit to 130 (=6.5*20)
             Change calc_rpm limit to 130 for 0 rpm detection
         On overflow, load timer 1 with 55536 (=655356-10000)
         Do movement detection at full rate, 100Hz
         Do a 3x moving average filter on movement to eliminate noise

     -> do a tidy up of valve events at BDC ?
      shorten false rev by overlapping valve events

gmot17
     cured false rev duration ambiguity when shaft moves, caused by hangover
interrupts
     optimised false rev numbers:
        dump_valve_delay = 35000, multiplier = 2 so total delay=70ms
        false_rev_delay  =  6000 (30% pulse overlap)
        switch off dump at m = 5 (25ms or so allowance for last valve to latch)

gmot18
    attempt to implement automatic rollback prediction, based on rollback at last
attempt.
    18a: works for both facing up and down hill

gmot19
    final full software for Danfoss visit to Denamark 25.11.01
    updated speed control loop parameters
    sorted dump valve operation

gmot19tT/S
    test version with data collection- constant update rate with timestamping
    T - torque mode data
    S - speed mode data

gmot20
    With data collection triggered by pot_bin[8], speed_override now obsolete
    Eliminated softpot2, used in Lucas mode for tacho input

*/


#include<stdio.h>
#include<80c196.h>

#pragma CCB(0xFA)
#pragma interrupt (timer1_overflow = 0)
#pragma interrupt (reset_timer = 4)
#pragma interrupt (fire_poppet = 5)

#define SERIAL_MODE 0x09   /* Mode 1 - 8-bit data asynchronous mode */
#define BAUD_LOW     0x08    /* 0x33 to set baud rate to 19200 for 16MHz, was 0x12 at
6MHz */
#define BAUD_HIGH   0x80   /* Select baud rate freq generator as timer1 */
#define bnc         0x08   /* Select pressure input from bnc */
#define REVERSE       1
#define FORWARD       0
#define TRUE          1
#define FALSE         0
#define cyl_disp    170   /* was 150, should be 170*/
#define false_rev_delay 6000
#define dump_valve_delay 35000 /* double this is dump valve delay in microsecs */
#define Max_count   10000 /* was 655536, now 10000 with 100Hz timer1 overflow */
```

```
#define buff_size 300
#define torque_control 1
#define speed_control 0
#define duty_control 2
#define stop 0
#define go 1
#define LPON 0x80
#define LPOFF 0x20
#define HPON 0x08
#define HPOFF 0x10


register int overflow_count;
register int offset, bnc_press, shaft_posn, last_shaft_posn, first_int, movement;
unsigned long one_rev_time;
register unsigned int new_time1,new_time2, new_time3 ,new_time;
register unsigned int count, is1, time1, pedal_dem, chan;
register unsigned int old_time1;
register unsigned char m, false_one, adj, once, pop_no;
register unsigned char start_offset_lp, start_offset_hp;
register unsigned char pop_enable, start_offset, int_count;
register unsigned char serial_enable, pot_change;
int speed_demand, torque_err, speed_err, dec;
unsigned int sample_count, old_rpm, rpm;
unsigned char last_pedal_dem, direction, dump_flag, dump_count, actual_dir, pop_temp,
status_count;
unsigned char lp_advance, hp_advance, last_lp_advance, last_hp_advance;

unsigned short ad(int),A,B,C,D,ten,thou,hun,temp_timer2,restart_count;
unsigned char pop_lp_on,pop_lp_off,pop_hp_on,pop_hp_off;
unsigned short pot_hp_on_time, pot_lp_on_time, pot_hp_off_time, pot_lp_off_time;
unsigned short lp_on_time, lp_off_time, hp_on_time, hp_off_time; /* Phase offsets for
valve events */
unsigned short last_lp_on_time, last_lp_off_time, last_hp_on_time, last_hp_off_time;
/* Phase offsets for valve events */

/* General purpose global temporary varaibles */
char temp_char;
int temp_int;
long temp_long;

/* Poppet address arrays */

const unsigned char lp_on[2][6]  = {{0x01, 0x02, 0x03, 0x04, 0x05, 0x06},
                                    {0x05, 0x04, 0x03, 0x02, 0x01, 0x06}};    /* Motoring
lp sequence  ON  */
const unsigned char lp_off[2][6] = {{0x04, 0x05, 0x06, 0x01, 0x02, 0x03},
                                    {0x02, 0x01, 0x06, 0x05, 0x04, 0x03}};    /* Motoring
lp sequence  OFF */
const unsigned char hp_off[2][6] = {{0x01, 0x02, 0x03, 0x04, 0x05, 0x06},    /* Shifted
left!!*/
                                    {0x05, 0x04, 0x03, 0x02, 0x01, 0x06}};   /* Motoring
hp sequence  OFF */
const unsigned char hp_on[2][6]  = {{0x04, 0x05, 0x06, 0x01, 0x02, 0x03},
                                    {0x02, 0x01, 0x06, 0x05, 0x04, 0x03}}; /* Motoring hp
sequence  ON  */
/*  Memory test  */
/*long mem[10];*/

/* Added for multiselector switch control */
short  last_pot_select, softpot1, softpot2, last_softpot1, last_softpot2, pot_select;
unsigned char pot_latch, restart_override;
short pot_bin[11], last_pot_bin[11];

/*Added for serial buffer */
unsigned short  add_buff, tx_buff;
unsigned char buff[buff_size];
char buff_overflow;

/* Added for  speed control loop and pressure command */
int speed_demand, torque_err, speed_err, dec, integ_err, last_speed_err;
int dspeed_err, av_torque, press_dem, speed_gain ,integ_gain, diff_gain;
long  full_torque_err;

/* Added for mode control */
char control_mode, speed_pedal_status, full_torque_start;
int brake_pedal, accel_pedal, pressure_in;
```

```
int speed_limit;

/* Added for porgressive start */
char start_count, torque_override;

/* Added for keeping track of poppet status */
char lp_status [7], hp_status[7],  pulse_save_override, full_start;
char check_pot_flag;

/* Added for self test */
unsigned char self_test, self_test_count, self_test_valve;

/* Added for hill start strategy */
int     last_rollback,     last_brake_pedal,     false_rev_posn,     false_rev_movement,
filter_movement, last_movement, total_rollback;
unsigned char pump_pedal, first_bdc, ready_for_encoder, overflow_divisor;

/* Data collection specials */
long timestamp;
char data_mode, relief;


void do_valve_event(poptype,number)
unsigned char poptype,number;
{

        ioport1 |= number;
        ioport1 |= poptype;
        ioport1 ^= poptype;
        ioport1 |= number;
        ioport1 |= number;
        ioport1 ^= number;


        if(poptype==LPON)
          lp_status[number] = 1;
        else if (poptype==LPOFF)
          lp_status[number] = 0;
        else if (poptype==HPON)
          hp_status[number] = 1;
        else if (poptype==HPOFF)
          hp_status[number] = 0;
}

void do_torque_override()
{
      torque_err = 100;
      full_torque_err = 100;

      if (control_mode == torque_control)
        press_dem = (accel_pedal/5) + 5;
      else
        press_dem = (speed_demand/7) + 60;

      if(press_dem <0)
        press_dem = -press_dem;

      if (press_dem > 220)
        press_dem = 220;


      pwm_control = press_dem;

}

void valve_status_print()
 {
   putchar (0x0d);
   putchar (0x09);
   putchar (0x09);
   putchar (0x09);
    putchar (0x4c);
    for (count=1; count<7; count++)
    {
      if (lp_status[count]  == 0)
          putchar (0x30);
      else if (lp_status[count] == 1)
```

```
            putchar (0x31);
        else
            putchar (0x58);
    }

    putchar (0x48);

    for (count=1; count<7; count++)
    {
      if (hp_status[count] == 0)
         putchar (0x30);
      else if (hp_status[count]  == 1)
          putchar (0x31);
      else
          putchar (0x58);
     }

}

/*
void buff_valve_status_print()
 {
    buff_putchar (0x0d);
    buff_putchar (0x09);
    buff_putchar (0x09);
    buff_putchar (0x09);
    buff_putchar (0x4c);
    for (count=1; count<7; count++)
    {
      if (lp_status[count]  == 0)
         buff_putchar (0x30);
      else if (lp_status[count] == 1)
          buff_putchar (0x31);
      else
          buff_putchar (0x58);
     }

     buff_putchar (0x48);

     for (count=1; count<7; count++)
    {
      if (hp_status[count] == 0)
         buff_putchar (0x30);
      else if (hp_status[count]  == 1)
          buff_putchar (0x31);
      else
          buff_putchar (0x58);
      }
 }
 */

 buff_putchar(char_out)
 char char_out;
 {
    add_buff++;

    if (add_buff == buff_size)
        add_buff = 0;

   if  (add_buff != tx_buff)
       buff[add_buff] = char_out;
   else
      putchar(0x21);
 }


 buff_transmit(number)
 int number;
 {
    while (number != 0)
    {
      if (tx_buff != add_buff)
      {
          tx_buff++;
          if (tx_buff == buff_size)
             tx_buff = 0;
```

```
        putchar(buff[tx_buff]);
      }
        number--;
    }
}


sample_ad()
{
   shaft_posn          = ad(3);
   pedal_dem           = ad(0);
   pot_select          = ad(5)/100;
   softpot1            = 1024-ad(6);
   softpot2            = 1024-ad(7);
   accel_pedal         = ad(1);
   brake_pedal         = ad(2);
   pressure_in         = ad(4)/4;
}


 void do_speed_control()
 {
     last_speed_err = speed_err;
     speed_err = speed_demand - rpm;

     integ_err += ((long)(integ_gain*speed_err)/50);

     if (integ_err > (int) 10000)
       integ_err = (int) 10000;
     if (integ_err < (int) -10000)
       integ_err = (int) -10000;

     dspeed_err = -(speed_err - last_speed_err);
    /* dspeed_err = 0;*/
    /* Positve speed_err = go faster */

     full_torque_err = ((long)speed_gain*speed_err)/100 + ((long)integ_err/100)
                                  + ((long)diff_gain*dspeed_err/100);

     /* For  Excel tests, was (speed_err*2) + (integ_err/100) + (dspeed_err/6) */

     /* 05.03.02 Altered constants here to get full span integ_err giving 100% torque
*/


     torque_err = full_torque_err;

     if (torque_err > 100)
        torque_err = 100;
     if (torque_err < -100)
        torque_err = -100;
 }

void do_speed_limit()
{
   if (rpm > speed_limit)
   {
     torque_err = torque_err - ((rpm - speed_limit)/2);

     if (torque_err > 100)
       torque_err = 100;
    if (torque_err < -100)
       torque_err = -100;
    }
}


measure_speed()
{
  shaft_posn = ad(3);

  last_movement = movement;
  movement = shaft_posn - last_shaft_posn;

  if (movement > 512 )            /* Presume rollover */
    movement = movement - 1024;
```

```
  else if (movement < -512)
    movement = 1024 + movement;

  last_shaft_posn = shaft_posn;



  filter_movement = (10*(movement + last_movement))/2;
  /* filter_movement is averaged over 2 samples to eliminate noise */
  /* (FIR digital filter) and multiplied by ten                    */

  /* Can determine direction from sampling analogue signal only below
     300rpm. Above that speed, assume it is the same as last time. */
  if ((rpm<300) || (ready_for_encoder == TRUE))
   {
     if (filter_movement > 0)
       {
          actual_dir = FORWARD;
          temp_long=filter_movement;
          temp_long = (long)(64 * temp_long) / 110;
          rpm=temp_long;
          /* for 65ms sample period was:        */
          /* rpm = (long) 100 * movement / 110; */

       }
     else
       {
         actual_dir = REVERSE;
         temp_long=filter_movement;
         temp_long = -(long)(64 * temp_long)/110;
         rpm=temp_long;

       }

     if (-6 < filter_movement && 6 > filter_movement)
       rpm = 0;
   }

}

check_restart_motoring()
{

  if (accel_pedal < 250)
  {
   pump_pedal = 1;
  }

  if (brake_pedal > 600 && last_brake_pedal < 600)
  {
    pump_pedal = 1;
    restart_count = 1;
  }

  last_brake_pedal = brake_pedal;

  if ( (restart_override == 1 || brake_pedal > 600) && (pump_pedal == 0) )
  {
    restart_count =1;
    return;
  }



  if ( (speed_pedal_status == stop && accel_pedal<100)
       && (rpm < 50)
     )
    poppets_off();



  if (rpm > 100 && direction==actual_dir)
  {
    restart_count=8;
    /*last_rollback=0;*/
```

```
  }

 /*
  if  ((direction != actual_dir) && (rpm > 20)&& (rpm <300))
      {
      once = 1;
      restart_count = 4;
}

*/

  if ( (rpm==0  || ((direction != actual_dir) && (rpm < 60)) )
  &&  ((accel_pedal > 100 || speed_pedal_status == go)) )/* Stopped but want to start
*/
    {
      if  (restart_count  ==  8  &&  restart_override  ==  0  &&  false_one  ==0  &&
ready_for_encoder==FALSE)
        {
          once = 1;   /* 'Once' flag is picked up in while loop to start FALSE rev */
          restart_count = 2;
          pump_pedal = 0;
        }

      restart_count++;

      if (restart_count > 8)  /* If motor stays stationary, retry FALSE rev */
        restart_count = 2;
    }
}

print_data()
{
  timestamp++;
  buff_print_int(timestamp);
  buff_putchar(0x2c);
  buff_print_int(speed_demand);
  buff_putchar(0x2c);
  buff_print_int(rpm);
  buff_putchar(0x2c);
  buff_print_int(integ_err);
  buff_putchar(0x2c);
  buff_print_int(dspeed_err);
  buff_putchar(0x2c);
  buff_print_int(torque_err);
  buff_putchar(0x2c);
  buff_print_int(full_torque_err);
  buff_putchar(0x2c);
  buff_print_int(press_dem);
  buff_putchar(0x2c);
  buff_print_int(pressure_in);
  buff_putchar(0x2c);

  if (relief==1)
    buff_putchar(0x31);
  else
    buff_putchar(0x30);

  buff_putchar(0x0d);

}


check_pots()
{

pot_change=1;

if (pot_select != last_pot_select)
  {
     pot_latch = 0;
     pot_change = 1;
   }

 if  ((pot_latch  ==  0)  &&  (softpot1  <  (pot_bin[pot_select]+20))  &&  (softpot1  >
(pot_bin[pot_select]-20)))
   {
     pot_latch = 1;
```

```
  }
if ( (softpot1 > last_softpot1 + 5) || (softpot1 <last_softpot1- 5) )
 {
  pot_change = 1;
  last_softpot1 = softpot1;
 }
/* if ( (softpot2 > last_softpot2 + 5) || (softpot2 <last_softpot2- 5) )
 {
  pot_change = 1;
  last_softpot2 = softpot2;
 }
 */
 if (pot_latch == 1 && pot_select != 0)
    pot_bin[pot_select] = softpot1;



 temp_char=1;
 while (temp_char < 11)
 {
     switch (temp_char)
     {
         case 1:
            pot_hp_on_time = pot_bin[temp_char]/6;
            break;
         case 2:
            pot_hp_off_time = pot_bin[temp_char]/3;
            break;
         case 3:
            pot_lp_on_time = pot_bin[temp_char]/6;
            break;
          case 4:
            pot_lp_off_time = pot_bin[temp_char]/3;
            break;
          case 5:
             speed_gain = pot_bin[temp_char]/10;
             break;
          case 6:
             diff_gain =  pot_bin[temp_char]/10;
             break;
          case 7:
             integ_gain = pot_bin[temp_char]/10;
             break;
          case 8:
             if (pot_bin[temp_char] > 500)
              data_mode = TRUE;
             else
             {
              data_mode = FALSE;
              timestamp=0;
             }
          case 9:
            speed_limit = pot_bin[temp_char] * 10;
          case 10:
            if (pot_bin[temp_char] > 512)
              serial_enable =1;
            else
              serial_enable =0;

            if (pot_bin[temp_char] >900)
              self_test = 1;
            else
              self_test = 0;

            if (pot_bin[temp_char] <100)
              self_test = 2;

            if (pot_bin[temp_char] <200)
              full_start = 0;
            else
              full_start = 1;

            break;
```

```
        }
        temp_char++;
    }


  if (data_mode == TRUE)
  {
   print_data();
   return;
  }


  if (pot_change == 1)
  {
     temp_char=1;
     buff_putchar(0x0D);

     while (temp_char < 11)
       {
          if (pot_select == temp_char)
          {
             if (pot_latch == 0 )
               {
                     if (softpot1 < pot_bin[pot_select])
                   buff_putchar(0x3e);
                 if (softpot1 > pot_bin[pot_select])
                   buff_putchar(0x3c);
                }
            else
               buff_putchar(0x2a);
           }
         else
             buff_putchar(0x20);

         buff_print_int(pot_bin[temp_char]);
         buff_putchar(0x20);
         temp_char++;
       }
    }


if (pot_change==1)
  {
  calc_timings();

    buff_putchar(0x09);
    buff_putchar(0x48);
    buff_putchar(0x47);
    buff_print_int(hp_on_time);
    buff_putchar(0x20);

    buff_putchar(0x48);
    buff_putchar(0x53);
    buff_print_int(hp_off_time);
    buff_putchar(0x20);

    buff_putchar(0x4C);
    buff_putchar(0x47);
    buff_print_int(lp_on_time);
    buff_putchar(0x20);

    buff_putchar(0x4C);
    buff_putchar(0x53);
    buff_print_int(lp_off_time);

    buff_putchar(0x20);
    buff_putchar(0x53);
    buff_print_int(rpm);

    buff_putchar(0x20);
    buff_putchar(0x44);
    buff_print_int(press_dem);

    buff_putchar(0x20);
    buff_putchar(0x50);
    buff_print_int(pressure_in);
```

```
    buff_putchar(0x20);
    buff_putchar(0x54);
    buff_print_int(full_torque_err);

    buff_putchar(0x20);
    buff_putchar(0x46);
    buff_print_int(torque_err);

    if (hp_advance == TRUE)
    {
     buff_putchar(0x48);
    }

    if (lp_advance == TRUE)
    {
     buff_putchar(0x4C);
    }

    buff_putchar(0x20);
    buff_putchar(0xbd);
    buff_print_int(softpot2);

    last_pot_select = pot_select;
    pot_change = 0;

    temp_char = 0;
    while (temp_char <10)
    {
       last_pot_bin[temp_char] = pot_bin[temp_char];
       temp_char++;
    }
      }
}

poppets_off()
{
   for (count=1; count<7; count++) /* resets monostables */
     {
      do_valve_event(LPOFF,count);
      do_valve_event(HPOFF,count);
     }



}

calc_timings()
{

/* Uses speed measurement "rpm" to advance timings */

        if (pot_lp_on_time > ((rpm/16)+18))
                lp_on_time = pot_lp_on_time - (rpm/16);
        else
                lp_on_time = 18;

        if (pot_lp_off_time > ((rpm/9)+7))
                lp_off_time = pot_lp_off_time - (rpm/9);  /* was 12*/
        else
                lp_off_time = 7;

        if (pot_hp_on_time > ((rpm/18)+20))
                hp_on_time = pot_hp_on_time - (rpm/18);
        else
                 hp_on_time = 20;

        if (pot_hp_off_time > ((rpm/7)+11))
                hp_off_time = pot_hp_off_time - (rpm/7);
        else
                hp_off_time = 11;

  if (hp_off_time >= 170)
    {
     if (hp_off_time > 180)
       {
       hp_off_time = hp_off_time - 170;
```

```
      hp_advance = FALSE;
      }
    else
      {
      if (hp_advance == TRUE)
         hp_off_time = 155;
      else
         {
         hp_off_time = hp_off_time - 170;
         }
      }
    }
 else
    hp_advance = TRUE;

 if (hp_off_time < 11)
   hp_off_time = 11;


 if (hp_off_time > 155)
   hp_off_time = 155;

 if ((rpm == 0) && (hp_advance==TRUE))
   {
    hp_advance = FALSE;
    hp_off_time = 5;
   }

 if (lp_off_time >= 170)
   {
    if (lp_off_time > 180)
      {
      lp_off_time = lp_off_time - 170;
      lp_advance = FALSE;
      }
    else
      {
      if (lp_advance == TRUE)
         lp_off_time = 163;
      else
         lp_off_time = lp_off_time - 170;
      }
    }
 else
    lp_advance = TRUE;

if (lp_off_time < 7)
   lp_off_time = 7;

if (lp_off_time > 163)
   lp_off_time = 163;

 if (torque_err < 0)
     {
      lp_off_time = 152;      /* LP ON when pumping */
     }

 if ((rpm == 0) && (lp_advance==TRUE))
   {
    lp_advance = FALSE;
    lp_off_time = 7;
   }

    /* switch off lpv left behind by advance */
    if ((lp_advance == TRUE )&&( last_lp_advance==FALSE))
       do_valve_event(LPOFF,lp_off[direction][0]);

    /* switch off hpv left behind by advance */
     if ((hp_advance == TRUE )&&( last_hp_advance==FALSE))
          do_valve_event(HPOFF,hp_off[direction][0]);

    last_lp_advance =  lp_advance;
    last_hp_advance = hp_advance;

}
```

```
void fire_poppet(void)
{
  if(m==9)
    return;

   if (dump_flag == TRUE) /* Special handling to double dump valve delay */
     {

      /* Set up timer interrupt for fire_poppet */
      new_time3 = timer1;
      hso_command = 0x019;
      hso_time = (new_time3 + dump_valve_delay);

      dump_count++;
      if (dump_count > 0)
        dump_flag=FALSE;

     return;
     }

   if ((m==0) && (dump_flag==FALSE) && (false_one == TRUE))  /* Check position after
dumping */
     {
       shaft_posn  = ad(3);
       get_offset();
       sector_posn();
       int_count=0;
       if(serial_enable==1)
       {
        putchar(0x40);
        print_int(shaft_posn);
        putchar(0x0d);
       }
     }

    new_time2 = timer2; /* position (incremented by LSB) */
    is1 = ios1;
    int_count++;
    if (int_count >= 10)
      int_count = 1;

    if (serial_enable == 1)
     {
       if (false_one == TRUE)
         putchar(0x2a);
       putchar(int_count + 0x30);
       putchar(0x20);
     }


/* Motoring Low Pressure valve power ON */

    if ((is1 & 0x01) == 0x01)
      {
       if (
          (false_one == FALSE)
          || ((m == 4) && (lp_off_time  > first_int))
          || ((m == 7) && (lp_on_time  < first_int))
          || m == 5
          || m == 6
          )
          {

           if ( (torque_err > 0) || false_one==TRUE)
             {

                do_valve_event(LPON,lp_on[direction][pop_lp_on + start_offset_lp]);

                          if (serial_enable == 1)
                          {
                              putchar(0x4C);
                              putchar(0x47);
                              putchar(0x30     +     lp_on[direction][pop_lp_on     +
start_offset_lp]);
                              putchar(0x20);
                                    putchar(0x30  +  pop_lp_on  +  start_offset_lp);
```

```
                        }

                }

            /* Actually switch LP OFF for active pumping*/

             if ( (dec < -100) && false_one==FALSE)     /*DISABLED!! idle trial*/
               {
                pop_temp = pop_lp_on;
                do_valve_event(LPOFF,lp_on[actual_dir][pop_temp]);

                dec += 100;

                 if (serial_enable == 1)
                   {
                     putchar(0x4C);
                     putchar(0x53);
                     putchar(0x30 + lp_on[actual_dir][pop_temp]);
                     putchar(0x20);
                           putchar(0x30 + pop_temp);
                    }


              }


           }
          pop_lp_on++;

          if (pop_lp_on <= 5 && false_one == FALSE)
            {
              hso_command = 0x058;
              hso_time = (cyl_disp*pop_lp_on + lp_on_time);
              /*hso_time = (new_time2 + cyl_disp); */
            }
          if ((pop_lp_on == 6) || ((pop_lp_on + start_offset_lp) == 6))
            {
              pop_lp_on = 0;
              start_offset_lp = 0;
            }

      }

/* Motoring Low Pressure valve power OFF */

     if ((is1 & 0x08) == 0x08)
       {

         if (1024==1024) /* MOD TO MOTOR ALL THE TIME */
           {
             if (lp_advance==TRUE)
               {
                if (pop_lp_off == 5)
                  pop_temp = 0;
                else
                  pop_temp = pop_lp_off + 1;
               }
             else
                pop_temp = pop_lp_off;

        dec += torque_err;
        if (dec > 1000)
          dec = 899;

        if (dec < -1000)
          dec = -899;

             if ( dec < 101)     /* LP ON for pumping or idling */

              {
                pop_temp = pop_lp_on - 2;
                if (pop_lp_on == 1)
                  pop_temp = 5;
                if (pop_lp_on == 0)
                  pop_temp = 4;

                do_valve_event(LPON,lp_off[actual_dir][pop_temp]);
```

```
                if (serial_enable == 1)
                {
                        putchar(0x4C);
                 putchar(0x47);
                        putchar(0x30 + lp_off[actual_dir][pop_temp]);
                        putchar(0x20);
                        putchar(0x30 + pop_temp);


                 }
              }

            if ( (false_one==TRUE) || (dec > 100) )                    /* Motoring  LP
OFF */
              {
               do_valve_event(LPOFF,lp_off[direction][pop_temp]);

               if (false_one==FALSE)
                     dec -= 100;


              if (serial_enable == 1)
                 {
                  putchar(0x4C);
                  putchar(0x53);
                  putchar(0x30 + lp_off[direction][pop_temp]);
                 putchar(0x20);
                 putchar(0x30 + pop_temp);
                  }
              }

        }


        pop_lp_off++;
        if (pop_lp_off <= 5 && false_one == FALSE)
        {
          hso_command = 0x05B;
          /* hso_time = (new_time2 + cyl_disp);  */
          hso_time = (pop_lp_off*cyl_disp + lp_off_time);
        }

        if (pop_lp_off == 6)
          pop_lp_off = 0;

        }


/* Motoring high pressure valve power on */

    if ((is1 & 0x02) == 0x02)
      {


       if (
          (false_one == FALSE)
          || ((m == 0) && (hp_off_time  > first_int))
          || m == 1
          || m == 2
          || ((m == 3) && (hp_on_time  < first_int))
          || (m == 3 && full_start == 1)
           )
         {

          if ( (false_one==TRUE) || (torque_err > 0)  )

                {

                 do_valve_event(HPON,hp_on[direction][pop_hp_on + start_offset_hp]);

                  if (serial_enable == 1)
                      {
                      putchar(0x48);
                      putchar(0x47);
                      putchar(0x30 + hp_on[direction][pop_hp_on + start_offset_hp]);
                      putchar(0x20);
```

```
                        putchar(0x30 + pop_hp_on + start_offset_hp);
                        }

        }


        }

      pop_hp_on++;

      if (pop_hp_on <= 5 && false_one == FALSE) /* set up timing for position based
timing */
        {
         hso_command = 0x059;
        /* hso_time = (new_time2 + cyl_disp);*/
        hso_time = (pop_hp_on*cyl_disp + hp_on_time);
        }
      if (pop_hp_on == 6 || (pop_hp_on + start_offset_hp) == 6)
        {
         pop_hp_on = 0;
         start_offset_hp = 0;
        }
     }

/* Motoring high pressure valve off */

    if ((isl & 0x04) == 0x04)
      {
       if (hp_advance==TRUE)
          {
            if (pop_hp_off == 5)
              pop_temp = 0;
            else
              pop_temp = pop_hp_off + 1;
          }
        else
          pop_temp = pop_hp_off;


       if (serial_enable == 1)
       {
              putchar(0x48);
              putchar(0x53);
              putchar(0x30 + hp_off[direction][pop_temp]);
              putchar(0x20);
              putchar(0x30 + pop_temp);
               }

         do_valve_event(HPOFF,hp_off[direction][pop_temp]);

        pop_hp_off++;
      if (pop_hp_off <= 5 && false_one == FALSE)
        {
         hso_command = 0x5A;
        /* hso_time = (new_time2 + cyl_disp); */
        hso_time = (pop_hp_off*cyl_disp + hp_off_time);
        }
      if (pop_hp_off == 6)
        pop_hp_off = 0;
    }



    /* INDEX SEQUENCE FOR FALSE REVOLUTION */
    if (false_one == TRUE)
      {
       if (m <= 7 )
         m++;
      }

   if(m==5)
   {
        ioport1 &=0xBF;  /* Switch off Dump Valve FET */
       putchar(0x0D);
        putchar(0x44);  /* Print `DS` */
        putchar(0x53);
```

```
    putchar(0x0D);
}


/* FLAG TO INDICATE STATIC REVOLUTION FINISHED */
if (m == 8)
  {


   temp_char = 1;
   while (temp_char < 7 && full_start==1)
   {
    if (lp_status[temp_char]==0 && hp_status[temp_char]==0)
      do_valve_event(LPON,temp_char);


    if (lp_status[temp_char]==1 && hp_status[temp_char]==1)
       do_valve_event(LPOFF,temp_char);

    temp_char++;
   }

   ready_for_encoder = TRUE;
   m=9;
   total_rollback=0;

}

/* POPPET INDEX SEQUENCE FORWARD */
if (direction == FORWARD)
  {
   if (pop_no == 5)
     pop_no = 0;
   else
     pop_no++;
  }

/* POPPET INDEX SEQUENCE REVERSE */
if (direction == REVERSE)
  {
   if (pop_no == 0)
     pop_no = 5;
   else
     pop_no--;
  }

/* SOFTWARE TIMER1 INTERRUPTS FOR STATIC REVOLUTION */
new_time3 = timer1;
if (m < 4  && false_one == TRUE)  /* 1st 3 hp valves on */
  {
   hso_command = 0x019;
   hso_time = (new_time3 + false_rev_delay);
  }

if (m >= 4 && m <= 7 && false_one == TRUE)  /* next 3 lp valves on */
  {
   hso_command = 0x018;
   hso_time = (new_time3 + false_rev_delay);
  }


if (serial_enable == 1)
{
  valve_status_print();
  putchar(0x20);
  putchar(0x40);
  temp_int=ad(3);
  print_int(temp_int);
  putchar(0x20);
  putchar(0x22);
  print_int(timer2);
  putchar(0x20);
  putchar(0x27);
  print_int(new_time3/10);
  putchar(0x20);
  putchar(0x69);
```

```
      print_int((new_time3 + false_rev_delay)/10);
      putchar(0x0D);

    }

    buff_transmit (4);

    return;

}

get_offset()  /* Determine offset position */
  {
    for (count = 0; count < 1021; count = count + 170)
      {
        if (count == 1020)
           adj = 4;
        else
           adj = 0;
        if (shaft_posn >= count && shaft_posn <= count + 170 + adj)
           offset = count / 170;
      }

    if (offset == 6)
      offset = 5;

    if (direction == REVERSE)
      offset = 5 - offset;
}

false_rev()  /* CREATE INITIAL POPPET FIRING SEQUENCE */
{
    dec = 729;
    false_one = TRUE;
    ready_for_encoder = FALSE;
    actual_dir = direction;
    once = pop_hp_on = pop_lp_on = pop_lp_off = pop_hp_off = 0;
    shaft_posn  = ad(3);

    false_rev_posn = shaft_posn;



    shaft_posn = shaft_posn + last_rollback;

    if (shaft_posn > 1024)
      shaft_posn = shaft_posn - 1024;
    if (shaft_posn < 0)
      shaft_posn = shaft_posn + 1024;



    get_offset();
    sector_posn();
    calc_timings();

    /*Ramp up start*/
    /*
    if (start_count==1)
      pwm_control = 70;
    if (start_count==2)
      pwm_control = 110;
    if (start_count==3)
      pwm_control = 150;
    if (start_count > 3)
      pwm_control = 200;
      */

    first_bdc = torque_override = 1;
    do_torque_override();

    start_count++;

  /*
    buff_putchar(0x0d);
    buff_putchar(0x50);
    buff_print_int(first_int);
```

```
  buff_putchar(0x20);
  buff_putchar(0x53);
  buff_putchar(0x30 + offset);
  buff_putchar(0x20);
  if (direction == FORWARD)
    buff_putchar(0x46);
  if (direction == REVERSE)
    buff_putchar(0x52);
  buff_putchar(0x0D); */

  putchar(0x0d);
  putchar(0x40);
  print_int(false_rev_posn);
  putchar(0x20);
  putchar(0x53);
  print_int(shaft_posn);
  putchar(0x50);
  print_int(first_int);
  putchar(0x20);
  putchar(0x53);
  putchar(0x30 + offset);
  putchar(0x20);
  if (direction == FORWARD)
    putchar(0x46);
  if (direction == REVERSE)
    putchar(0x52);
  putchar(0x0D);

  /* Print out:
    @(shaft_posn) P(first_int) S(offset) F/R
  */

  int_count = 0;

  if (offset < 3)
    start_offset = offset + 3;
  else
    start_offset = offset - 3;

  start_offset_hp = start_offset;
  start_offset_lp = start_offset;


 ioport1 |= 0x40;  /* switch on Dump Valve FET  */


 /* Set up timer interrupt for fire_poppet */
 ioc2 = 0x80;                    /* Clear any software timers */
 new_time3 = timer1;
 hso_command = 0x019;
 hso_time = (new_time3 + dump_valve_delay);
 dump_flag=1;
 dump_count=0;

 putchar(0x44);    /* Print `DG` */
 putchar(0x47);
 putchar(0x0D);

 poppets_off();


}

print_tabs(tabcount)
char tabcount;
{
  temp_char =0;
  while (temp_char <=tabcount)
  {
    putchar(0x09);
    temp_char++;
  }
}
```

```
do_self_test()
{
 if (self_test == 1)  /* Pressure test with flow through open valves*/
 {
  switch (self_test_count)
  {
    case 1:
      putchar(0x0D);
      putchar(0x54);
      putchar(0x65);
      putchar(0x73);
      putchar(0x74);
      putchar(0x31);
      putchar(0x0d);
      print_tabs(self_test_valve);
      ioport2 |= 0x80;
      putchar(0x44);    /* Print `DG` */
      putchar(0x47);
      putchar(0x0D);
      /* fire dump*/
      pwm_control = 0;
      /* set pump press = 0*/
      poppets_off();
      /* switch off all poppets */
      break;
    case 25:
      do_valve_event(LPON,self_test_valve);
      print_tabs(self_test_valve);
      putchar(0x4C);
      putchar(0x47);
      putchar(0x30 + self_test_valve);
      putchar(0x0D);

      /* fire LPV , report*/
      break;
    case 35:
      do_valve_event(HPON,self_test_valve);
      print_tabs(self_test_valve);
      putchar(0x48);
      putchar(0x47);
      putchar(0x30 + self_test_valve);
      putchar(0x0D);

      /* fire HPV, report */
      break;
    case 45:
      ioport2 &=0x7F;
      /* stop dump */
      pwm_control = 50;
      /* set pressure = 50 */
      break;
    case 100:
      sample_ad();
      print_tabs(self_test_valve);
      print_int(pressure_in);
      putchar(0x0D);

      /* sample pressure, report to serial */

      do_valve_event(LPOFF,self_test_valve);

      print_tabs(self_test_valve);
      putchar(0x4C);
      putchar(0x53);
      putchar(0x30 + self_test_valve);
      putchar(0x0D);

      /* release LPV */
      break;
    case 150:
      sample_ad();
      print_tabs(self_test_valve);
      print_int(pressure_in);
      putchar(0x0D);
      /* sample pressure, report */
      poppets_off();
```

```
      /* release HPV */
      break;
    case 151:
      self_test_valve++;
      if (self_test_valve == 7)
       self_test_valve =1;
      /* increment valve */
      self_test_count = 0;

      /* set self_test_count = 0*/
      break;
   }
   self_test_count++;
}

if (self_test ==2)   /* Cycle quickly through valves for audible clicks */
 {
 switch (self_test_count)
 {/*
   case 1:
     putchar(0x0D);
     putchar(0x54);
     putchar(0x65);
     putchar(0x73);
     putchar(0x74);
     putchar(0x32);

     sample_ad();
     putchar(0x20);
     putchar(0x41);
     print_int(accel_pedal);
     putchar(0x20);

     putchar(0x42);
     print_int(brake_pedal);

     putchar(0x0d);
     print_tabs(self_test_valve);
     ioport1 |= 0x40;
     putchar(0x44);
     putchar(0x47);
     putchar(0x0D);

     pwm_control = 0;

     poppets_off();

     break;
    case 5:
     do_valve_event(LPON,self_test_valve);

     print_tabs(self_test_valve);
     putchar(0x4C);
     putchar(0x47);
     putchar(0x30 + self_test_valve);
     putchar(0x0D);


     break;
    case 10:
     do_valve_event(HPON,self_test_valve);
     print_tabs(self_test_valve);
     putchar(0x48);
     putchar(0x47);
     putchar(0x30 + self_test_valve);
     putchar(0x0D);


     break;

    case 15:

     do_valve_event(LPOFF,self_test_valve);
     print_tabs(self_test_valve);
     putchar(0x4C);
     putchar(0x53);
     putchar(0x30 + self_test_valve);
```

```
      putchar(0x0D);

      break;
    case 20:
      sample_ad();
      print_tabs(self_test_valve);
      print_int(pressure_in);
      putchar(0x0D);

      poppets_off();

      break;
    case 21:
      self_test_valve++;
      if (self_test_valve == 7)
       self_test_valve =1;

      self_test_count = 0;

      break;*/
   }
   self_test_count++;

  }

}


void timer1_overflow(void)
{
   /*  ------------------------------------------- */
   /*  Do the following every overflow_count = 100Hz */
   /*  ------------------------------------------- */


 if((false_one==TRUE) && (ready_for_encoder==FALSE))
   return;

    /* window select register allows timer1 to be altered*/
    wsr=0x0f;
    timer1 = 55535;
    wsr=0x00;

    measure_speed();

    if ( (false_one == TRUE) && (ready_for_encoder == TRUE) )
     {

       /*putchar(0x4d);
       print_int(movement);*/

          putchar(0x46);
          print_int(filter_movement);

          total_rollback=total_rollback+filter_movement;

          if (  ((total_rollback<-5000) && (direction==FORWARD))
              ||((total_rollback>5000) && (direction==REVERSE))
             )
            {
            poppets_off();
            putchar(0x2a);
            putchar(0x50);
            putchar(0x4F);
            putchar(0x2a);
            }
            /* failed start going up hill*/


       if ( (actual_dir == direction) || (rpm==0) )
        {
           change_to_encoder();
           return;
        }

       if ((actual_dir != direction) && (rpm>600))
       {
```

```
        poppets_off();
        }

      }



    if (overflow_count < 130)     /* Used by rpm measurement, stops at 20 till reset at
bdc */
         overflow_count++;
    else
      rpm = 0;     /* Make sure rpm gets updated when motor stopped */

    if(overflow_divisor < 6)
      {
       overflow_divisor++;
         if ((rpm < 500))
             buff_transmit(13);
       return;
      }
    else
      overflow_divisor = 0;
/*
putchar(0x4d);
print_int(movement);
putchar(0x72);
print_int(rpm);
putchar(0x66);
print_int(filter_movement);
putchar(0x20);
*/

  /*  ------------------------------------------------- */
  /* Do the following only every 6 overflows = 16.67Hz       */
  /*  ------------------------------------------------- */

  if (self_test!=0)
   {
     do_self_test();
     return;
   }

  demand();

  do_speed_control();
  do_torque_control();

  if ( (rpm > 100 && direction == actual_dir)
      || ((brake_pedal > 250 && accel_pedal < 100) && control_mode == torque_control)
      || (speed_pedal_status==stop && control_mode == speed_control)
     )
  {
    torque_override = 0;
    start_count = 1;
  }


  /*
  if (torque_override==1)
   putchar(0x54);
  else
   putchar(0x58);
   */



   if (false_one == FALSE)
       check_restart_motoring();


     /* check for updated timings due to speed or pot change and print out */

   if (((status_count/2)*2==status_count) && (false_one == FALSE))
     {
        if (rpm < 300)
        {
```

```
            sample_ad();
            check_pots();
          }
          else
             check_pot_flag = 1;
      }


    if (once == 1)
      {
        demand();
        false_rev();
        once = 0;
      }

    status_count++;
    if (status_count > 7)
    {
      status_count = 0;
    }



}

void calc_rpm()
{
    /* Recalibrated for 100Hz overflows */

    /* One overflow of timer1 is 65 ms */
     if (overflow_count > 0)
     {
       temp_long=Max_count;
       one_rev_time    =    (long)    ((65535    -    old_time1)+((overflow_count-1)    *
temp_long)+(new_time1-55535));
     }
     else
       one_rev_time = (new_time1 - old_time1);

     if (overflow_count < 129)
       rpm  =  (long)  (5998800  /  (one_rev_time/10));  /*  5998800  calibrated  to  0.2%
accuracy with 16MHz*/
     else
       rpm = 0;


     /*putchar(0x4e);
     print_int(overflow_count);
     putchar(0x52);
     print_int(rpm);
     putchar(0x0d);*/

 }

do_torque_control()
{

 if (control_mode == torque_control)
 {
  if (brake_pedal <250)
       {
       torque_err=accel_pedal/9;
       }
  else
    torque_err=-(4*(brake_pedal-250))/30;
  }

    if (torque_err > 100)
       torque_err = 100;
    if (torque_err < -100)
       torque_err = -100;

     do_speed_limit();

     full_torque_err = torque_err;

     torque_err = (200 * full_torque_err/ (pressure_in+1)) ;
```

```
      temp_int= ((torque_err  * pressure_in )/100);

      if(temp_int<0)
        temp_int=-temp_int;

      if (temp_int < 20)
        temp_int = 19;

      if (temp_int > 200)
        temp_int = 200;

      press_dem = temp_int;


     if (torque_err > 100)
        torque_err = 100;
     if (torque_err < -100)
        torque_err = -100;

     if (torque_override == 1)
     {
       do_torque_override();
     }


     pwm_control = press_dem;    /* Pressure demand */

     if ( (pressure_in > ((6*press_dem)/5 + 10)) && (torque_err < 0) )
     {
       ioport2 |=0x80;
       relief=TRUE;
     }
     else
     {
       ioport2 &=0x7f;
       relief=FALSE;
     }
     /*
     print_int(full_torque_err);
     putchar(0x20);
     putchar(0x44);
     print_int(press_dem);*/

}

/*
void do_displacement_control()
{
   if (brake_pedal <250)

     if (1==1)
      torque_err=accel_pedal/10;
     else
       torque_err=100;
  else
     torque_err=-(brake_pedal/10);

         if (torque_err > 100)
        torque_err = 100;
     if (torque_err < -100)
        torque_err = -100;
 }
*/


void reset_timer(void)
{

if (false_one == TRUE)    /* gate out reset_timer during false rev, stops crashes */
  return;


  temp_timer2 = timer2;
  timer2 = 0;

  /* print out value of timer2 at first bdc after false rev to show position errors */
```

```
  if (first_bdc==1)
  {
   putchar(0x0D);
   putchar(0x54);
   print_int(temp_timer2);
   putchar(0x0D);
   first_bdc = 0;
  }

  new_time2 = timer2;
  old_time1 = new_time1;
  new_time1 = timer1; /* real time timer */

  calc_rpm();
  calc_timings();

   if ((rpm > old_rpm + 15)|| (rpm < old_rpm - 15))
    {
     old_rpm = rpm;
     pot_change = 1;
    }

  overflow_count = 0;
  int_count = 0;
  offset = 0;

  sample_ad();


  ioc2 = 0x80;                    /* Clear any software timers */

  if (false_one == FALSE)
    {

     hso_command = 0x58;
     hso_time = (new_time2 + lp_on_time);    /* Set timing for LP ON */

     hso_command = 0x59;
     hso_time = (new_time2 + hp_on_time);    /* Set timing for HP ON */

     hso_command = 0x5A;
     hso_time = (new_time2 + hp_off_time);     /* Set timing for HP OFF */

     hso_command = 0x5B;
     hso_time = (new_time2 + lp_off_time);    /*    Set timing for LP OFF */
    }
  offset = 0;
  pop_no = 0;

  pop_lp_on = pop_lp_off = pop_hp_on = pop_hp_off = 0;



    if (serial_enable == 1)
     {
       putchar(0x42);
                putchar(0x44);
       putchar(0x43);
       putchar(0x0d);
       putchar(0x0A);
     }

     if (check_pot_flag ==1)
       check_pots();

     check_pot_flag = 0;

}


demand() /* Determine demanded direction of travel */
{

 if (pedal_dem >= 470)
   {
     speed_pedal_status = go;
     direction = FORWARD;
```

```
     speed_demand = (long )    ((pedal_dem - 469)*4);
    }
 else if (pedal_dem <= 440)
    {
      speed_pedal_status = go;
      direction = REVERSE;
      speed_demand = (long) ((441-pedal_dem)*4);
     }

 else if (pedal_dem <470 && pedal_dem >440)
    {
       speed_pedal_status = stop;
       speed_demand = 0;
     }

  if (speed_demand > speed_limit-50)
       speed_demand = speed_limit-50;

  if (control_mode == speed_control && rpm == 0 && speed_pedal_status == stop
       && accel_pedal > 100 )
       control_mode = torque_control;

  if (control_mode == torque_control && rpm < 100)
  {
     if((ioport2 & 0x04) == 0x04)
       direction = FORWARD;
    else
       direction = REVERSE;

    if (rpm==0 && speed_pedal_status == go && brake_pedal < 250)
      control_mode = speed_control;
   }

  if (control_mode == speed_control && brake_pedal > 250)
    control_mode = torque_control;

}

sector_posn() /* Determines position 0-170 in any sector */
{
 if (direction == FORWARD)
   first_int = (shaft_posn - (offset * 170));

 if (direction == REVERSE)
   first_int = (1024 - shaft_posn - (offset * 170));
}


change_to_encoder()
 /* Change to encoder, set first interrupts for when motor first moves
    Replaces set_int */
{
     start_offset = start_offset_lp = start_offset_hp = m = 0;

     /* print out false rev position Bxxx */

    shaft_posn=ad(3);

    false_rev_movement = shaft_posn - false_rev_posn;

    if (false_rev_movement > 512 )            /* Presume rollover */
      false_rev_movement = false_rev_movement - 1024;
    else if (false_rev_movement < -512)
      false_rev_movement = 1024 + false_rev_movement;


    last_rollback=total_rollback/10;

    if (last_rollback >500)
      last_rollback = 100;

    if (last_rollback <-500)
      last_rollback = -100;

    get_offset();
    sector_posn();
    if (direction == FORWARD)
```

```
   timer2 = shaft_posn;
else
   timer2 = 1024-shaft_posn;

new_time2 = timer2;

pop_hp_on = pop_hp_off = pop_lp_on = pop_lp_off = offset;

ioc2=0x80;  /* Connect encoder to timer */

if (first_int < hp_off_time)
  time1 = hp_off_time - first_int + new_time2;
else
  {
  time1 = hp_off_time + cyl_disp - first_int + new_time2;
  pop_hp_off = offset + 1;
  if (pop_hp_off == 6)
    pop_hp_off = 0;
  }
hso_command = 0x5A;
hso_time = time1;  /* Set timing for HP OFF */

if (first_int < lp_off_time)
  time1 = lp_off_time - first_int + new_time2;
else
  {
  time1 = lp_off_time + cyl_disp - first_int + new_time2;
  pop_lp_off = offset + 1;
  if (pop_lp_off == 6)
    pop_lp_off = 0;
  }
hso_command = 0x5B;
hso_time = time1 ; /*   Set timing for LP OFF */

if (first_int < hp_on_time)
  time1 = hp_on_time - first_int + new_time2;
else
  {
  time1 = hp_on_time + cyl_disp - first_int + new_time2;
  pop_hp_on = offset + 1;
  if (pop_hp_on == 6)
    pop_hp_on = 0;
  }
hso_command = 0x59;
hso_time = time1 ; /*   Set timing for HP ON */

if (first_int < lp_on_time)
  time1 = lp_on_time - first_int + new_time2;
else
  {
  time1 = lp_on_time + cyl_disp  - first_int + new_time2;
  pop_lp_on = offset + 1;
  if (pop_lp_on == 6)
     pop_lp_on = 0;
  }
hso_command = 0x58;
hso_time = time1 ; /*   Set timing for LP ON */

 if (serial_enable == 1)   /* 'E' */
     {
       buff_putchar(0x45);
       buff_putchar(0x0D);
     }

false_one = FALSE;
ready_for_encoder = FALSE;

putchar(0x0d);
putchar(0x42);
print_int(false_rev_posn);
putchar(0x20);
putchar(0x55);
print_int(shaft_posn);
putchar(0x20);
putchar(0x4d);
print_int(false_rev_movement);
putchar(0x20);
```

```
    putchar(0x3c);
    print_int(last_rollback);
    putchar(0x20);
    putchar(0x52);
    print_int(total_rollback/10);
    putchar(0x20);

}


unsigned short ad(channel)    /* Sample analogue channels */
int channel;
{
   ad_command = 0x008+channel;
   while( (ad_result_lo & 0x008) != 0x008 );
   while( (ad_result_lo & 0x008) == 0x008 );
   return((ad_result_hi&0xFF)*0x004)+((ad_result_lo/0x040)&0x003);
}


print_int(value) /* Print to screen unbuffered integers > 10 */
int value;
{
   if (value < 0)
     {
      putchar(0x2D);
      value = (int)(0 - value);
     }

   D = value/1000;
   putchar(D+48);

   thou = 1000*D;
   C = (value-thou)/100;
   putchar(C+48);

   hun = 100*C;
   B = (value-thou-hun)/10;
   putchar(B+48);

   ten = 10*B;
   A = value-thou-hun-ten;
   putchar(A+48);

}

buff_print_int(value) /* Print to serial buffer integers > 10 */
int value;
{
   if (value < 0)
     {
      buff_putchar(0x2D);
      value = (int)(0 - value);
     }

   D = value/1000;
   buff_putchar(D+48);

   thou = 1000*D;
   C = (value-thou)/100;
   buff_putchar(C+48);

   hun = 100*C;
   B = (value-thou-hun)/10;
   buff_putchar(B+48);

   ten = 10*B;
   A = value-thou-hun-ten;
   buff_putchar(A+48);

}


main()
{
/* This initialisation code is executed once, immediately on power-up*/
```

```
    ioc1 = 0x025;    /* Timer1 overflow  & TXD output */
    sp_con = SERIAL_MODE;
    baud_rate = BAUD_LOW;
    baud_rate = BAUD_HIGH;
    int_mask = 0x031;     /* External HSI.0 & Software timer  */
                          /* Timer overflow interrupts */
    init_putchar();
    pop_no = 0;
    overflow_count = 0;
    direction = FORWARD;
    status_count = 0;
    m = 0;
    false_one = 0;
    ready_for_encoder=FALSE;

    ioport1 =0;
    pop_no = 0;
    pop_hp_on = 0;
    pop_lp_on = 0;
    pop_lp_off = 0;
    pop_hp_off = 0;
    int_count = 0;
    hp_advance = FALSE;
    lp_advance = FALSE;
    sample_count = 0;
    pwm_control = 0;
    control_mode = torque_control;

    self_test_valve = 1;
    self_test_count = 0;
    self_test = 0;
    filter_movement = 0;

    /* Default value of pot controlled variables */
    pot_bin[1] = 99  * 6; /* HG */
    pot_bin[2] = 200 * 3; /* HS */
    pot_bin[3] = 120   * 6; /* LG */
    pot_bin[4] = 213 * 3; /* LS */
    pot_bin[5] = 150; /* Speed loop: Speed gain % */
    pot_bin[6] = 0; /* Speed loop: Differential gain % */
    pot_bin[7] = 500; /* Speed loop: Integral gain */
    pot_bin[8] = 450;       /* if > 500 then go into data collection mode*/
    pot_bin[9] = 130;  /* speed limit is ten times this*/
    pot_bin[10] = 450; /* if >500 then serial enable of all events */

    tx_buff=0;
    add_buff=0;
    timestamp=0;

    /* Temp for diagnostics */
    serial_enable = 0; /* if == 1 then serial record of every valve event */
    restart_override = 0;  /* if == 1 then automatic restart is inhibited */
    pulse_save_override =1; /* if ==0 then duplicate lp pulses are avoided */
    once = 0;               /* TEST ONLY */ /* if ==1 then do a false rev on reset or
powerup */
    full_start = 1;  /* if ==1 then make sure each cylinder has either LPV or HPV on
                        at end of false rev */
    start_count = 1;
    torque_override = 0;
    sample_ad();
    last_shaft_posn = shaft_posn;
/* Set up initial phase offsets for valve events */

   calc_timings();

    putchar(0x0D);
    putchar(0x40);
    print_int(shaft_posn); /* prints DAC value to RS232 */
    putchar(0x0D);
    check_pots();

    ioc0 = 0x80;           /* was 0xA8 */
    ioc1 = 0x2D;     /* Timer1 overflow  & TXD output was 0x29 */

    new_time2 = timer2 = shaft_posn;    /* analogue position signal */

    int_pending = 0;
```

```
    enable();

    poppets_off(); /* Reset all poppets */

    init_pwm0();


    while(1) /* main loop */
      {

      /* Loop waiting for next interrupt */

      }
}
```

9.8 DDP propel demonstrator - system controller software

```
/* GLUCAS software for Lucas motor mode
(c) 2002-2003 Niall Caldwell, Artemis Intelligent Power Ltd.

GLUCAS01

Based on gmot20
Eliminated fire_poppet, false_rev, change_to_encoder, get_offset, sector_posn....
100Hz fundamental heartbeat on timer1_overflow with divisor of 6 for check_pots
speed_demand is now 0 to 1 million
Send out flow demand to PWM based on foot pedal position

GLUCAS02
Get tacho input from Jon board, polarity is tacho_direction
Switch direction and dump valves for direction changeover

GLUCA03
Added speed feedback to accelerator control, gain set in pot_bin[1]
Dump valve now active whenever speed_demand=0

=>Connect a digital input to a switch on the dash to change between speed and torque
modes

GLUCA04
Mods after first road trials
Rollback dump valve switch off
Filter on speed pedal
Rescaled brake pedal

GLUCA05
Delays implemented for displacement change in timer1_overflow

GLUCA06
First attempt at positon control using encoder

GLUCA06t
test program lists values of all analog and digital inputs

GLUCAS07
includes test program if potbin[6] >900, now tests FET outputs too.

GLUCAS08
New routines for reading digital inputs and addressing LED FETs

GLUCAS09
New mode logic, scaling, filtering, for use with new control box.

GLUCAS10
Bipolar pedal_speed_demand gives smoother speed control thru zero

GLUCAS11
Leakage detection at startup

GLUCAS12
Verbose serial log of actions, written to vbuff_putchar(), vbuff_printint()
Suppress F/R led switching every cycle
Added motor override mode

GLUCAS13 As for Leif Tandrup (Sauer Danfoss) demo

GLUCAS14 With new ram mode override mode. Spare switch selects whether to wait for
pressure to fall before
switching off dump valve.
Only switch to full displacement when pressure > 50 bar.
Only allow scale changes in idle mode

GLUCAS15 For NVTF visit. Removed pressure condition for disp change.
Longer filters for loose mode.
Nov 2003 - new logic to prevent jerks during disp changes, recalibrated brake pedal.


*/
#include<stdio.h>
```

```
#include<80c196.h>

#pragma CCB(0xFA)
#pragma interrupt (timer1_overflow = 0)
#pragma interrupt (reset_timer = 4)
#pragma interrupt (fire_poppet = 5)

#define SERIAL_MODE 0x09   /* Mode 1 - 8-bit data asynchronous mode */
#define BAUD_LOW     0x08    /* 0x33 to set baud rate to 19200 for 16MHz, was 0x12 at
6MHz */
#define BAUD_HIGH   0x80   /* Select baud rate freq generator as timer1 */
#define bnc         0x08   /* Select pressure input from bnc */
#define REVERSE       1
#define FORWARD       0
#define TRUE         1
#define FALSE        0
#define Max_count   10000 /* was 655536, now 10000 with 100Hz timer1 overflow */
#define buff_size 400
#define accel_control 1
#define speed_control 0
#define position_control 2
#define idle 3
#define stop 0
#define go 1
#define ON 1
#define OFF 0
#define LPON 0x80
#define LPOFF 0x20
#define HPON 0x08
#define HPOFF 0x10
#define full 1
#define half 0

#define spare_LED 4
#define override_LED 5
#define position_LED 8
#define speed_LED 9
#define acceleration_LED 10
#define idle_LED 11
#define forward_LED 12
#define reverse_LED 13
#define buzzer 6

register int overflow_count;
register int offset, bnc_press, first_int;
unsigned long one_rev_time;
register unsigned int new_time1,new_time2, new_time3 ,new_time;
register unsigned int count, is1, time1, pedal_dem, chan;
register unsigned int old_time1;
register unsigned char m, false_one, adj, once, pop_no;
register unsigned char start_offset_lp, start_offset_hp;
register unsigned char pop_enable, start_offset;
register unsigned char serial_enable, pot_change;
int speed_err, dec;
unsigned int old_rpm, rpm;
unsigned char last_pedal_dem, direction, dump_flag, actual_dir, pop_temp;


unsigned short ad(int),A,B,C,D,ten,thou,hun,temp_timer2,restart_count;

/* General purpose global temporary varaibles */
char temp_char;
int temp_int;
long temp_long;

/* Added for multiselector switch control */
short  last_pot_select, softpot1, softpot2, last_softpot1, last_softpot2, pot_select;
unsigned char pot_latch, restart_override;
short pot_bin[11], last_pot_bin[11];

/*Added for serial buffer */
unsigned int  add_buff, tx_buff;
unsigned char buff[buff_size];
char buff_overflow;

/* Added for  speed control loop and pressure command */
int  speed_err, dec, integ_err, last_speed_err;
```

```
int dspeed_err, av_torque;

/* Added for mode control */
char control_mode, speed_pedal_status, full_torque_start;
int brake_pedal, accel_pedal, pressure_in;
int speed_limit;

/* Added for hill start strategy */
int     last_rollback,     last_brake_pedal,     false_rev_posn,     false_rev_movement,
last_movement, total_rollback;
unsigned    char    pump_pedal,    first_bdc,    ready_for_encoder,    overflow_divisor,
overflow_divisor2;

/* Data collection specials */
long timestamp;
char data_mode, relief, verbose;

char disp_valve, dump_valve, dump_timeout, dir_valve, tacho_direction, tacho_deadband;
long speed_demand, pedal_speed_demand, accel_speed_demand, accel_demand, flow_demand,
filter_pedal_speed_demand;
int tacho, flow_output, accel_speed_feedback, speed_pedal_range;
char disp_off_timer, disp_on_timer, low_to_high_delay, high_to_low_delay, input_mode,
last_control_mode;
int movement, shaft_posn, last_shaft_posn;
long filter_movement, position_speed_demand, position_err;
unsigned char test_count,dump_token,actual_disp;

/* Control Box */
char switch_scale, switch_filters, switch_override, switch_direction, switch_spare;
char pos_gain, pos_filter, heartbeat, dir_LED;

/* leak detect */
int leak_count, leak_time, pulse_count, pulse_time, pulse_flow;
unsigned long leak_speed;
char ram_mode;

void do_valve_event(poptype,number)
unsigned char poptype,number;

{
        /* 0x40 must be set so it can used as an input */

        /* ioport1 |= (number & 0x40);
         ioport1 |= (poptype & 0x40);
         ioport1 &= ~poptype;
         ioport1 |= (number & 0x40);
         ioport1 |= (number & 0x40);
         ioport1 &= ~number;*/

          ioport1 |= number;
         ioport1 |= number;
         ioport1 |= poptype;
          ioport1 |= poptype;
         ioport1 &= ~poptype;
         ioport1 |= number;
         ioport1 |= number;
         ioport1 &= ~number;
         ioport1 |= 0x40;


}

poppets_off()
{
   for (count=1; count<7; count++) /* resets monostables */
     {
      do_valve_event(LPOFF,count);
      do_valve_event(HPOFF,count);
     }
}

void switch_dir_valve(status)
unsigned char status;
{

  if (status==ON)
    {
```

```
    do_valve_event(LPON,3);
    dir_valve=ON;
    vbuff_putchar(0x44);
    vbuff_putchar(0x52);
    vbuff_putchar(0x31);
    vbuff_putchar(0x20); /* DR1 */
   }
  else
   {
    do_valve_event(LPOFF,3);
    dir_valve=OFF;
    vbuff_putchar(0x44);
    vbuff_putchar(0x52);
    vbuff_putchar(0x30);
    vbuff_putchar(0x20); /* DR0 */
   }
}

void switch_disp_valve(status)
unsigned char status;
{

  if (status==ON)
   {
    do_valve_event(LPON,1);
    disp_valve=ON;
    vbuff_putchar(0x44);
    vbuff_putchar(0x53);
    vbuff_putchar(0x31);
    vbuff_putchar(0x20); /* DS1 */
   }
  else
   {
    do_valve_event(LPOFF,1);
    vbuff_putchar(0x44);
    vbuff_putchar(0x53);
    vbuff_putchar(0x30);
    vbuff_putchar(0x20); /* DS0 */
    disp_valve=OFF;
   }
}

void switch_dump_valve(status)
unsigned char status;
{

  if (status==ON)
   {
    do_valve_event(LPON,2);
    dump_valve=ON;
    vbuff_putchar(0x44);
    vbuff_putchar(0x4D);
    vbuff_putchar(0x31);
    vbuff_putchar(0x20); /* DM1 */
    }
  else
   {
    do_valve_event(LPOFF,2);
    dump_valve=OFF;
    vbuff_putchar(0x44);
    vbuff_putchar(0x4D);
    vbuff_putchar(0x30);
    vbuff_putchar(0x20); /* DM0 */
   }
}

switch_all_valves(status)
unsigned char status;
{
  switch_dump_valve(status);
  switch_dir_valve(status);
  switch_disp_valve(status);
}

switch_LED(id,status)
unsigned char id, status;
{
```

```
  if (status==ON)
  {
    switch(id)
    {
      case spare_LED:
        do_valve_event(LPON,4);
        break;
      case override_LED:
        do_valve_event(LPON,5);
        break;
      case buzzer:
        do_valve_event(LPON,6);
        break;
      case position_LED:
        do_valve_event(HPON,1);
        break;
      case speed_LED:
        do_valve_event(HPON,2);
        break;
      case acceleration_LED:
        do_valve_event(HPON,3);
        break;
      case idle_LED:
        do_valve_event(HPON,4);
        break;
      case forward_LED:
        do_valve_event(HPON,5);
        break;
      case reverse_LED:
        do_valve_event(HPON,6);
        break;
    }
  }
  if (status==OFF)
  {
    switch (id)
    {
      case spare_LED:
        do_valve_event(LPOFF,4);
        break;
      case override_LED:
        do_valve_event(LPOFF,5);
        break;
      case buzzer:
        do_valve_event(LPOFF,6);
        break;
      case position_LED:
        do_valve_event(HPOFF,1);
        break;
      case speed_LED:
        do_valve_event(HPOFF,2);
        break;
      case acceleration_LED:
        do_valve_event(HPOFF,3);
        break;
      case idle_LED:
        do_valve_event(HPOFF,4);
        break;
      case forward_LED:
        do_valve_event(HPOFF,5);
        break;
      case reverse_LED:
        do_valve_event(HPOFF,6);
        break;

    }
  }
}

switch_all_LEDs(status)
unsigned char status;
{
  temp_char=4;
  while (temp_char<14)
  {
      switch_LED(temp_char,status);
```

```
      temp_char++;
  }
}


Read_Control_Switches()
{
 /* NB return codes:
   ON=switch down  */

      /* Scale Factor, P1.6 , blue */

 if (control_mode==idle)  /*only allow range changing in idle mode*/
 {
      if ((ioport1 & 0x40) ==0x40)
        switch_scale = OFF;
      else
        switch_scale = ON;
 }

      /* Override, P2.2 , black */
      if ((ioport2 & 0x04) ==0x04)
         switch_override = OFF;
      else
         switch_override = ON;

       /* Filters, P2.3, yellow */
      if ((ioport2 & 0x08) ==0x08)
         switch_filters = OFF;
      else
         switch_filters = ON;

      /* Spare switch, P2.6 , violet */
      if ((ioport2 & 0x40) ==0x40)
         switch_spare = OFF;
      else
         switch_spare = ON;

       /* Direction switch, P2.7 , grey */
      if ((ioport2 & 0x80) ==0x80)
          switch_direction = ON;
      else
          switch_direction = OFF;

 }


 static unsigned rootfind2(unsigned long val)
 /* Finds the integer square root of a long */
 /* From www.azillionmonkeys.com/qed/sqroot.html */
 {
  unsigned long temp, g=0, b=0x8000, bshft=15;
  do {
        if (val >= (temp = (((g<<1)+b)<<bshft--)))
          {
            g+=b;
            val-=temp;
          }
     } while (b>>=1);

    return g;
 }


 buff_putchar(char_out)
 char char_out;
 {
    add_buff++;

    if (add_buff == buff_size)
        add_buff = 0;

   if  (add_buff != tx_buff)
       buff[add_buff] = char_out;
   else
      putchar(0x21);
 }
```

```
vbuff_putchar(char_out)        /* Only prints if in verbose mode */
char char_out;
{
   if(verbose==FALSE)
      return;

   add_buff++;

   if (add_buff == buff_size)
       add_buff = 0;

  if  (add_buff != tx_buff)
      buff[add_buff] = char_out;
  else
     putchar(0x21);
}



buff_transmit(number)
int number;
{
   while (number != 0)
   {
     if (tx_buff != add_buff)
     {
         tx_buff++;
         if (tx_buff == buff_size)
            tx_buff = 0;
         putchar(buff[tx_buff]);
      }
        number--;
    }
}


sample_ad()
{
   shaft_posn          = ad(3);  /* encoder for position control */
   pedal_dem           = ad(0);  /* heel/toe speed control pedal input */
   pot_select          = ad(5)/100;
   softpot1            = 1024-ad(6);
   tacho               = ad(7);  /* Was softpot2, rewire to tacho input */
   accel_pedal         = ad(1);
   brake_pedal         = ad(2);
   pressure_in         = ad(4)/4;
}


do_accel_control()
{
 if (control_mode != accel_control)
   return;

  accel_demand=0;
  if (brake_pedal <350)
      {
      accel_demand=accel_pedal;  /* max accel 950 */
      }
  else
 {
    accel_demand=-3*(brake_pedal-350)/2;        /* max deccel 750 */
    if (brake_pedal>800)
       accel_demand-=(brake_pedal-800)*4;      /* emergency stop */
 }
    /* feedback of  speed to give some speed control to pedal */

  if(switch_filters==ON)
    temp_long = ((long)((accel_speed_feedback/10)*tacho)/(50))+200;
  else
    {
      temp_long = ((long)((accel_speed_feedback/5)*tacho)/(50))+400;
      if (accel_demand>0)
         accel_demand=accel_demand*2;
    }
    accel_demand-=temp_long;
```

```
    if (accel_demand < 0)
    {
      accel_demand = accel_demand/2;
    }

    /* Gentle decceleration if direction switch flipped while driving */
    if ((switch_direction!=dir_valve) && (accel_demand > -200))
      accel_demand = -200;

  /* accel_demand is acceleration demand */
  /* 1000 means 0 to full speed_demand (1e6) in 1 second */

    accel_speed_demand += accel_demand*5;

    if(accel_speed_demand<0)
      accel_speed_demand=0;
    if (accel_speed_demand>1000000)
      accel_speed_demand=1000000;

    /* avoid error buildup if torque too high to switch to low motor displacement*/
    if (accel_speed_demand>550000 && disp_valve==OFF)
      accel_speed_demand=550000;



}

print_data()
{
  timestamp++;
  buff_print_int(timestamp);
  buff_putchar(0x2c);


  if (input_mode==TRUE)  /* Special test mode to examine analig and digital inputs and
test FETS */
  {

      /* Analog inputs */
      temp_int =0;
      while (temp_int<8)
        {
          buff_print_int(ad(temp_int));
          buff_putchar(0x2c);
          temp_int++;
        }


      /* Digital inputs connected to switches */

      /* P1.6 , blue */
      if ((ioport1 & 0x40) ==0x40)
          buff_putchar(0x31);
      else
          buff_putchar(0x30);
       buff_putchar(0x2c);

      /* P2.2 , black */
      if ((ioport2 & 0x04) ==0x04)
          buff_putchar(0x31);
      else
          buff_putchar(0x30);
       buff_putchar(0x2c);

       /* P2.3 , yellow */
      if ((ioport2 & 0x08) ==0x08)
          buff_putchar(0x31);
      else
          buff_putchar(0x30);
       buff_putchar(0x2c);

      /* P2.6 , grey */
      if ((ioport2 & 0x40) ==0x40)
          buff_putchar(0x31);
      else
          buff_putchar(0x30);
```

```
        buff_putchar(0x2c);

        /* P2.7 , violet */
       if ((ioport2 & 0x80) ==0x80)
           buff_putchar(0x31);
       else
           buff_putchar(0x30);
        buff_putchar(0x2c);

       buff_putchar(0x0d);

    if(overflow_divisor2==0)
     {

       if (test_count>23)
          test_count = 0;

       /* FETS off at test_count = 1,3,5,..... */
       /* HPON 0,2,4,6,8,10
          LPON 12,14,16,18,20,22 */


       if ( (test_count/2)*2 != test_count)
          poppets_off();
       else
          {
           if (test_count < 12)
             do_valve_event(LPON,(test_count/2)+1);
           else
             do_valve_event(HPON,(test_count/2)-5);
           }

       test_count++;
     }

  }

else /* General data output of transmission variables */

    {

      buff_print_int(tacho);

      buff_putchar(0x2c);
      temp_long=speed_demand/1000;
      buff_print_int(temp_long);

      buff_putchar(0x2c);
      buff_print_int(pedal_dem);

      buff_putchar(0x2c);
      buff_print_int(accel_pedal);

      buff_putchar(0x2c);
      buff_print_int(brake_pedal);


      buff_putchar(0x2c);
      buff_print_int(flow_demand/1000);

      /*
      buff_putchar(0x2c);
      buff_print_int(integ_err);
      buff_putchar(0x2c);
      buff_print_int(dspeed_err);
      buff_putchar(0x2c);
      */
      buff_putchar(0x2c);
      buff_print_int(accel_demand);

      buff_putchar(0x2c);
      buff_print_int(pressure_in);

      buff_putchar(0x2c);
      buff_print_int(shaft_posn);

      buff_putchar(0x2c);
```

```
    buff_print_int(filter_movement/10);

    buff_putchar(0x2c);
    buff_print_int(position_err/100);


    buff_putchar(0x2c);
    if (control_mode==speed_control)
      buff_putchar(0x73);
    if (control_mode==position_control)
      buff_putchar(0x70);
    if (control_mode==accel_control)
      buff_putchar(0x61);
    if (control_mode==idle)
      buff_putchar(0x69);

    buff_putchar(0x2c);
    if (dir_valve==ON)
      buff_putchar(0x31);
    else
      buff_putchar(0x30);
    buff_putchar(0x2c);

    if (disp_valve==ON)
      buff_putchar(0x31);
    else
      buff_putchar(0x30);

     buff_putchar(0x2c);
    if (dump_valve==ON)
      buff_putchar(0x31);
    else
      buff_putchar(0x30);


      buff_putchar(0x2c);
    if (tacho_direction==FORWARD)
      buff_putchar(0x31);
    else
      buff_putchar(0x30);

    buff_putchar(0x0d);

  }


}


check_pots()
{
if (pot_select != last_pot_select)
  {
     pot_latch = 0;
     pot_change = 1;
   }

 if  ((pot_latch  ==  0)  &&  (softpot1  <  (pot_bin[pot_select]+20))  &&  (softpot1  >
(pot_bin[pot_select]-20)))
   {
     pot_latch = 1;
    }

  if ( (softpot1 > last_softpot1 + 5) || (softpot1 <last_softpot1- 5) )
   {
    pot_change = 1;
    last_softpot1 = softpot1;
   }

   if (pot_latch == 1 && pot_select != 0)
      pot_bin[pot_select] = softpot1;

   temp_char=1;
   while (temp_char < 11)
   {
```

```
    switch (temp_char)
    {
        case 1:
           tacho_deadband = pot_bin[temp_char]/10;
           break;
        case 2:
           accel_speed_feedback = pot_bin[temp_char];
           break;
        case 3:
           low_to_high_delay = pot_bin[temp_char]/10;
           break;
         case 4:
           high_to_low_delay = pot_bin[temp_char]/10;
           break;
         case 5:
           speed_pedal_range = pot_bin[temp_char]/10;
            break;
         case 6:
            leak_time =  pot_bin[temp_char]/10;
            break;
         case 7:
            pulse_time = pot_bin[temp_char]/10;
         case 8:
            pulse_flow = pot_bin[temp_char];
            break;
         case 9:
            leak_speed = (long) pot_bin[temp_char]*100;
             break;
         case 10:
            if (pot_bin[temp_char] > 500)
              data_mode = TRUE;
            else
            {
              data_mode = FALSE;
              timestamp=0;
              test_count=0;
            }

            if (pot_bin[temp_char] >900)
              input_mode = TRUE;
            else
              input_mode = FALSE;

            if (pot_bin[temp_char] <200)
              verbose=TRUE;
            else
              verbose=FALSE;

            break;

     }
     temp_char++;
 }

if (data_mode==TRUE)
  return;

if (pot_change == 1)
{
   temp_char=1;
   buff_putchar(0x0D);

   while (temp_char < 11)
     {
        if (pot_select == temp_char)
        {
           if (pot_latch == 0 )
             {
               if (softpot1 < pot_bin[pot_select])
                 buff_putchar(0x3e);
               if (softpot1 > pot_bin[pot_select])
                 buff_putchar(0x3c);
             }
          else
             buff_putchar(0x2a);
        }
       else
```

```
            buff_putchar(0x20);

          buff_print_int(pot_bin[temp_char]);
          buff_putchar(0x20);
          temp_char++;
      }


    buff_putchar(0x09);
    buff_print_int(tacho_deadband);
    buff_putchar(0x20);

    buff_putchar(0x20);
    buff_putchar(0x53);
    buff_print_int(rpm);

    buff_putchar(0x20);
    buff_putchar(0x50);
    buff_print_int(pressure_in);

    buff_putchar(0x0d);


    last_pot_select = pot_select;
    pot_change = 0;

    temp_char = 0;
    while (temp_char <10)
    {
       last_pot_bin[temp_char] = pot_bin[temp_char];
       temp_char++;
    }

  }


}

print_tabs(tabcount)
char tabcount;
{
  temp_char =0;
  while (temp_char <=tabcount)
  {
    putchar(0x09);
    temp_char++;
  }
}

void handwheel_movement()
{

  last_movement = movement;
  movement = shaft_posn - last_shaft_posn;

  if (movement > 512 )            /* Presume rollover */
    movement = movement - 955;

  else if (movement < -512)
    movement = 955 + movement;

  last_shaft_posn = shaft_posn;

}

void handwheel_filter()
{
  /* swop polarity of movement so fowards is forwards*/
  temp_long = ((pos_filter-1)*filter_movement - 10*movement)/pos_filter;
  filter_movement = temp_long;

}

do_position_control()
{

   if (control_mode != position_control)
```

```
    return;

  if (switch_filters==ON)
    {
     pos_gain = 2;        /* was 3,20*/
     pos_filter = 40;
    }
  else
    {
     pos_gain = 10;      /* was 20,3*/
     pos_filter = 5;
    }

  handwheel_filter();

if (-1000000<position_err && position_err< 1000000)
{
  if(switch_scale==ON)
    position_err += 4*(filter_movement/10);
  else
    position_err += 20*(filter_movement/10);
}


   if (position_err > 0)
   {
     position_speed_demand = position_err*pos_gain;
     direction=FORWARD;
     position_err -= position_speed_demand/200;
   }
   else
   {
     position_speed_demand = -position_err*pos_gain;
     direction=REVERSE;
     position_err -= -position_speed_demand/200;
   }


}


void timer1_overflow(void)
{
   /*  ---------------------------------------------- */
   /*  Do the following every overflow_count = 100Hz */
   /*  ---------------------------------------------- */

    /* window select register allows timer1 to be altered*/
    wsr=0x0f;
    timer1 = 55535;
    wsr=0x00;



    sample_ad();
    Read_Control_Switches();
    Read_Control_Lever();
    Decide_Control_Mode();
    Update_Mode_LED();
    do_position_control();
    do_accel_control();


    if (control_mode==speed_control)
    {
      if (filter_pedal_speed_demand > 0)
      {
        direction = FORWARD;
        speed_demand = filter_pedal_speed_demand;
      }
      if (filter_pedal_speed_demand < 0)
      {
        direction = REVERSE;
        speed_demand = -filter_pedal_speed_demand;
      }
      if (filter_pedal_speed_demand == 0)
```

```
        speed_demand = 0;
    }
    else if (control_mode==accel_control)
      {
       if (switch_scale==ON)
        speed_demand = accel_speed_demand/4;
       else
         speed_demand = accel_speed_demand;
       }
    else if (control_mode==position_control)
      speed_demand = position_speed_demand;
    else if (control_mode==idle)
      speed_demand = 0;

   if (speed_demand >1000000)
     speed_demand = 1000000;          /*Full flow with low displacement motor */
   if (speed_demand < 0)
     speed_demand = 0;



     if (direction==REVERSE && dir_valve==OFF)
      {
       speed_demand = 0;
       if (tacho_direction==REVERSE || tacho < tacho_deadband)
        {
         if (dump_valve==OFF)
         {
           switch_dump_valve(ON);
           dump_timeout=30;
         }
         if (dump_token>0)
           dump_token--;
         if   (((pressure_in   <   20   ||   (ram_mode==TRUE   &&   switch_spare==ON))   &&
dump_token==0) || dump_timeout==0)
          {
           switch_dir_valve(ON);
           switch_dump_valve(OFF);
           dump_token=3;
          }
        }
      }

     if (direction==FORWARD && dir_valve==ON)
      {
       speed_demand = 0;
       if (tacho_direction==FORWARD || tacho < tacho_deadband)
        {
         if (dump_valve==OFF)
           {
           switch_dump_valve(ON);
           dump_timeout=30;
           }

         if   (pressure_in   <   20   ||   dump_timeout==0   ||(switch_spare==ON   &&
ram_mode==TRUE))
          {
           switch_dir_valve(OFF);
           switch_dump_valve(OFF);
          }

        }
      }

    if (dump_timeout>0)
      dump_timeout--;

    if (dump_timeout==1)
       switch_dump_valve(OFF);

    /* only switch to small displacement if you are fast enough, you want to go fast,
      and you have enough torque to accelerate */

    if (tacho > 370 && disp_valve==OFF && speed_demand > 540000 && pressure_in < 120
&& ram_mode==FALSE)
      {
       /* decision to change to half displacement */
```

```
        switch_disp_valve(ON);


        /* Use constant delay */
        disp_on_timer = low_to_high_delay;
     }

    if (tacho < 330 && disp_valve==ON && speed_demand<480000 /*&& pressure_in > 50*/)
    {
        /* decision to change to full displacement */
        switch_disp_valve(OFF);
       /*Use constant delay*/
           disp_off_timer = high_to_low_delay;
    }


 /*         if   ((disp_valve==ON   &&   disp_on_timer==0)   ||   (disp_valve==OFF   &&
disp_off_timer>0) )

       actual_disp=half;
    else

      actual_disp=full;
*/
     if  (actual_disp==full  &&  disp_valve==ON  &&  (disp_on_timer==0  &&  pressure_in
>120))
        actual_disp=half;

     if( actual_disp==half && disp_valve==OFF && (disp_off_timer==0 || pressure_in <
40))
        actual_disp=full;

     if (actual_disp==full)
         flow_demand=(speed_demand*20)/11;
     else
         flow_demand=speed_demand;

     if (disp_off_timer > 0)
       disp_off_timer--;

    if (disp_on_timer > 0)
       disp_on_timer--;

    /* Detect leakage problem at startup
       Motor is leaking if you are commanding a speed, your are stationary or going
backwards,
       and the direction valve is correclty set. If this persists for leak_count,
intiate a pulse start */

    if (speed_demand>leak_speed && (tacho<10 /*|| tacho_direction!=direction*/) &&
       ((direction==FORWARD && dir_valve==OFF)||(direction==REVERSE && dir_valve==ON))
        && pulse_count==0 && ram_mode==FALSE)
      {
       if (leak_count==0)
         {
          vbuff_putchar(0x6c);
          vbuff_putchar(0x73);
          vbuff_putchar(0x20); /* ls */
         }

       leak_count++;
       if (leak_count==leak_time)
       {
          leak_count=0;
          pulse_count=pulse_time;
          vbuff_putchar(0x70);
          vbuff_putchar(0x73);
          vbuff_putchar(0x20); /* ps */
       }
      }
      else
       leak_count=0;  /* leakage has to persist for leak_time to initiate a flow pulse
*/

    if (pulse_count>0)
    {
      pulse_count--;
```

```
      flow_demand=pulse_flow*1000;
      if (tacho>tacho_deadband && tacho_direction==direction)
        pulse_count=0;

      if (pulse_count==0)
      {
       vbuff_putchar(0x70);
       vbuff_putchar(0x65);
       vbuff_putchar(0x20); /* pe */
      }
    }

    /* Use spare switch to disable leak detection */

    if (switch_override==ON && ram_mode==FALSE)
    {
      ram_mode=TRUE;
      switch_LED(override_LED,ON);
      vbuff_putchar(0x52);
      vbuff_putchar(0x4d);
      vbuff_putchar(0x20);
    }
    else if (switch_override==OFF && ram_mode==TRUE)
    {
      ram_mode=FALSE;
      switch_LED(override_LED,OFF);
      vbuff_putchar(0x72);
      vbuff_putchar(0x6d);
      vbuff_putchar(0x20);
    }



    flow_output=rootfind2(flow_demand)*10/39;
    if (flow_output>255)
      flow_output=255;
    pwm_control=flow_output;

     if (heartbeat==90)
    {
      switch_LED(spare_LED,ON);
    }

    if (heartbeat>99)
     {
      switch_LED(spare_LED,OFF);
      heartbeat=0;
     }

    heartbeat++;

    buff_transmit(10);



    if(overflow_divisor < 10)
      {
       overflow_divisor++;
       return;
      }
    else
      overflow_divisor = 0;

  /*  ------------------------------------------------ */
  /* Do the following only every 10 overflows = 10Hz       */
  /*  ------------------------------------------------ */


  handwheel_movement();   /* Measure movement of handwheel in last 100ms */

  if (data_mode == TRUE)
    print_data();



/*     if(overflow_divisor2 ==1)
```

```
   do_valve_event(LPON,3);
  if(overflow_divisor2==3)
       do_valve_event(LPOFF,3);*/

   if (overflow_divisor2 < 5)
     {
       overflow_divisor2++;
       return;
     }
   else
     overflow_divisor2 = 0;



   /*  ------------------------------------------------- */
  /* Do the following only every 50 overflows = 2Hz      */
  /*  ------------------------------------------------  */

     check_pots();


}


Read_Control_Lever() /* Determine demanded direction of travel and calculate speed
demand */
{

/* p_s_d & f_p_s_d now bipolar, direction control based on f_p_s_d polarity */

 temp_long=pedal_dem;

 if (temp_long >= 560)
   {
     speed_pedal_status = go;
     pedal_speed_demand = (long ) ((temp_long - 560)*2590) ;
    }
 else if (temp_long <= 450)
    {
      speed_pedal_status = go;
      pedal_speed_demand = (long) ((temp_long-450)*2590);
     }

 else if (pedal_dem <561 && pedal_dem >449)
    {
       speed_pedal_status = stop;
       pedal_speed_demand = 0;
     }

    /* Scale speed demand lever for low gear only */
   if (switch_scale==ON)
      pedal_speed_demand=pedal_speed_demand*speed_pedal_range/100;

     /* Filter pedal input for smoother response */
  if (switch_filters==ON)
    filter_pedal_speed_demand        =        (filter_pedal_speed_demand*199        +
pedal_speed_demand)/200;
  else
   /*       filter_pedal_speed_demand       =       (filter_pedal_speed_demand*9       +
pedal_speed_demand)/10;*/
    filter_pedal_speed_demand = pedal_speed_demand;


}


Decide_control_mode()
{

  /* Decide whether to switch into idle */

  if (control_mode == speed_control && tacho < tacho_deadband
      &&    filter_pedal_speed_demand<3000    &&    filter_pedal_speed_demand>-3000    &&
speed_pedal_status == stop)
    {
       control_mode = idle;
    }
```

```
  if    (control_mode    ==    accel_control    &&    tacho    <    tacho_deadband    &&
accel_speed_demand==0)
    {
       control_mode = idle;
    }

  if (control_mode == position_control && position_err<300 && position_err >-300 &&
movement >-4 && movement <4)
    control_mode=idle;


  /* While in idle mode do these things: */

  if (control_mode==idle && ram_mode==FALSE)
  {
      if(switch_direction==ON)
       {
         direction = REVERSE;
         if(dir_LED != REVERSE)
         {
           switch_LED(forward_LED,OFF);
           switch_LED(reverse_LED,ON);
           dir_LED=REVERSE;
         }
       }
      else
       {
         direction = FORWARD;
         if (dir_LED != FORWARD)
         {
           switch_LED(reverse_LED,OFF);
           switch_LED(forward_LED,ON);
           dir_LED=FORWARD;
         }
       }
  }

  /* Decide whether to switch out of idle */

  if (control_mode==idle && accel_pedal > 50)
    control_mode=accel_control;

  if (control_mode==idle && speed_pedal_status==go)
    control_mode=speed_control;

  if (control_mode==idle && (movement > 4 || movement <-4) )
  {
    control_mode = position_control;
    position_err = 0;
  }

  /* Brake override to jump into accel_control from any mode if brake pedal pressed*/
  if  ( ((control_mode  ==  speed_control)  ||  (control_mode  ==  position_control))  &&
brake_pedal > 300)
    {
      control_mode = accel_control;

     if (switch_scale==ON)
        accel_speed_demand = speed_demand*5;
       else
         accel_speed_demand = speed_demand;

    }

}


Update_Mode_LED()
{
  if (control_mode != last_control_mode)
  {
      switch_LED(acceleration_LED,OFF);
      switch_LED(speed_LED,OFF);
      switch_LED(position_LED,OFF);
      switch_LED(idle_LED,OFF);
```

```
        switch(control_mode)
        {
           case accel_control:
             switch_LED(acceleration_LED,ON);
             break;
           case speed_control:
             switch_LED(speed_LED,ON);
             break;
           case position_control:
             switch_LED(position_LED,ON);
             break;
           case idle:
             switch_LED(idle_LED,ON);
             break;
        }
     }
     last_control_mode=control_mode;
}


unsigned short ad(channel)    /* Sample analogue channels */
int channel;
{
    ad_command = 0x008+channel;
    while( (ad_result_lo & 0x008) != 0x008 );
    while( (ad_result_lo & 0x008) == 0x008 );
    return((ad_result_hi&0xFF)*0x004)+((ad_result_lo/0x040)&0x003);
}


print_int(value) /* Print to screen unbuffered integers > 10 */
int value;
{
    if (value < 0)
      {
       putchar(0x2D);
       value = (int)(0 - value);
      }

    D = value/1000;
    putchar(D+48);

    thou = 1000*D;
    C = (value-thou)/100;
    putchar(C+48);

    hun = 100*C;
    B = (value-thou-hun)/10;
    putchar(B+48);

    ten = 10*B;
    A = value-thou-hun-ten;
    putchar(A+48);

}

buff_print_int(value) /* Print to serial buffer integers > 10 */
int value;
{
    if (value < 0)
      {
       buff_putchar(0x2D);
       value = (int)(0 - value);
      }

    D = value/1000;
    buff_putchar(D+48);

    thou = 1000*D;
    C = (value-thou)/100;
    buff_putchar(C+48);

    hun = 100*C;
    B = (value-thou-hun)/10;
    buff_putchar(B+48);
```

```
   ten = 10*B;
   A = value-thou-hun-ten;
   buff_putchar(A+48);

}

vbuff_print_int(value) /* Print to serial buffer integers > 10 if in verbose mode*/
int value;
{
   if (verbose==FALSE)
     return;

   if (value < 0)
     {
      buff_putchar(0x2D);
      value = (int)(0 - value);
     }

   D = value/1000;
   buff_putchar(D+48);

   thou = 1000*D;
   C = (value-thou)/100;
   buff_putchar(C+48);

   hun = 100*C;
   B = (value-thou-hun)/10;
   buff_putchar(B+48);

   ten = 10*B;
   A = value-thou-hun-ten;
   buff_putchar(A+48);

}


void reset_timer(void)
{
}

main()
{
/* This initiliasation code is ececuted once at power-up */
   iocl = 0x025;    /* Timer1 overflow  & TXD output */
   sp_con = SERIAL_MODE;
   baud_rate = BAUD_LOW;
   baud_rate = BAUD_HIGH;
   int_mask = 0x031;     /* External HSI.0 & Software timer  */
                         /* Timer overflow interrupts */
   init_putchar();

   overflow_count = 0;
   direction = FORWARD;

   pwm_control = 0;
   control_mode = accel_control;
   filter_pedal_speed_demand = 0;
   actual_disp=full;

      /* Default value of pot controlled variables */
   pot_bin[1] = 200; /* Tacho deadband is this / 10 */
   pot_bin[2] = 200; /* Accel speed feedback */
   pot_bin[3] = 100; /* Delay ms low to high gear */
   pot_bin[4] = 70; /* Delay ms high to low gear */
   pot_bin[5] = 200; /* Low speed range of pedal, 1000 = full speed */
   pot_bin[6] = 600; /* leak_time in ms*/
   pot_bin[7] = 100; /* pulse_time in ms*/
   pot_bin[8] = 500; /* pulse_flow  */
   pot_bin[9] = 400;  /* leak_speed is this times 100*/
   pot_bin[10] = 450; /* if >500 then serial enable of all events */

   tx_buff=0;
   add_buff=0;
   timestamp=0;
   heartbeat=0;

   accel_demand=0;
```

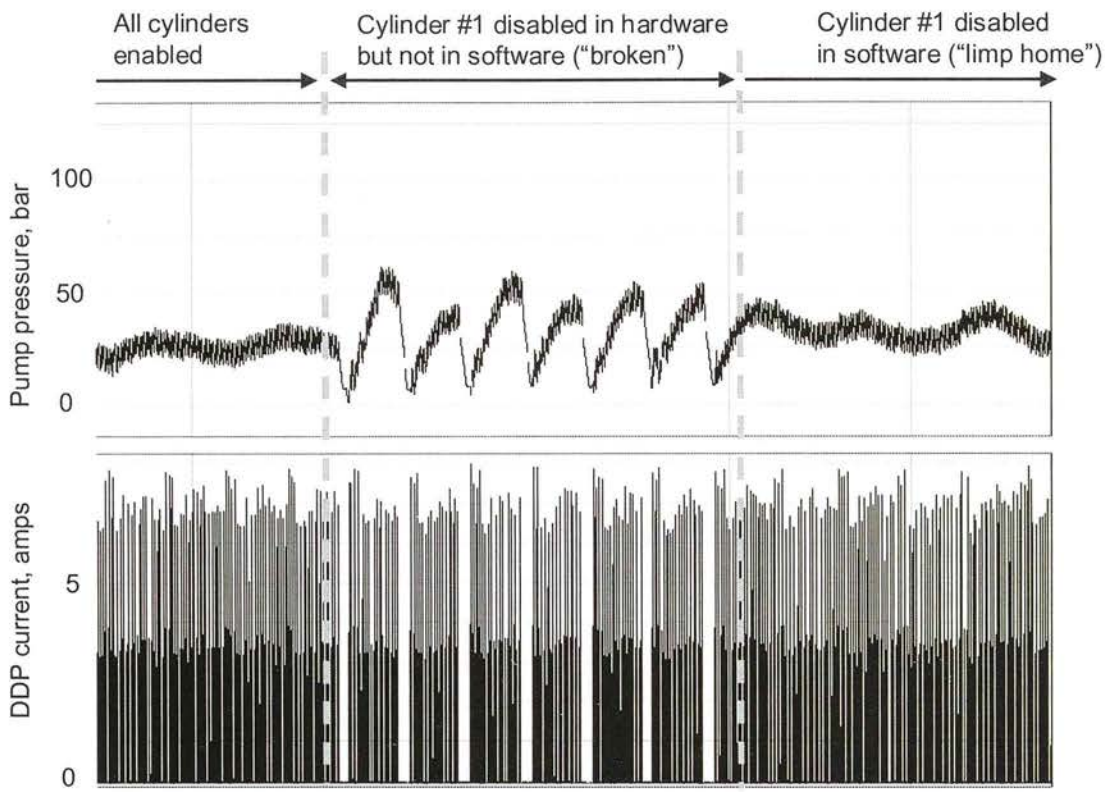
```
    speed_demand=0;
    filter_movement=0;

    pulse_count=leak_count=0;

    /* Temp for diagnostics */
    serial_enable = 0; /* if == 1 then serial record of every valve event */


    sample_ad();

    ioc0 = 0x80;           /* was 0xA8 */
    ioc1 = 0x2D;     /* Timer1 overflow  & TXD output was 0x29 */

    ioport1 |= 0x40; /* must be set so it can be used as an input */
    enable();
    init_pwm0();

    poppets_off();
    switch_all_valves(OFF);
    switch_all_LEDs(OFF);
    last_control_mode=99;
    control_mode=idle;
    dump_token=2;
    dir_LED=2;
    verbose=FALSE;
    ram_mode=FALSE;


   putchar(0x52);

   while(1) /* main loop */
     {

     /* Loop waiting for next interrupt*/

     }
}
```

9.9 Effect of a broken cylinder on DDP flow quality

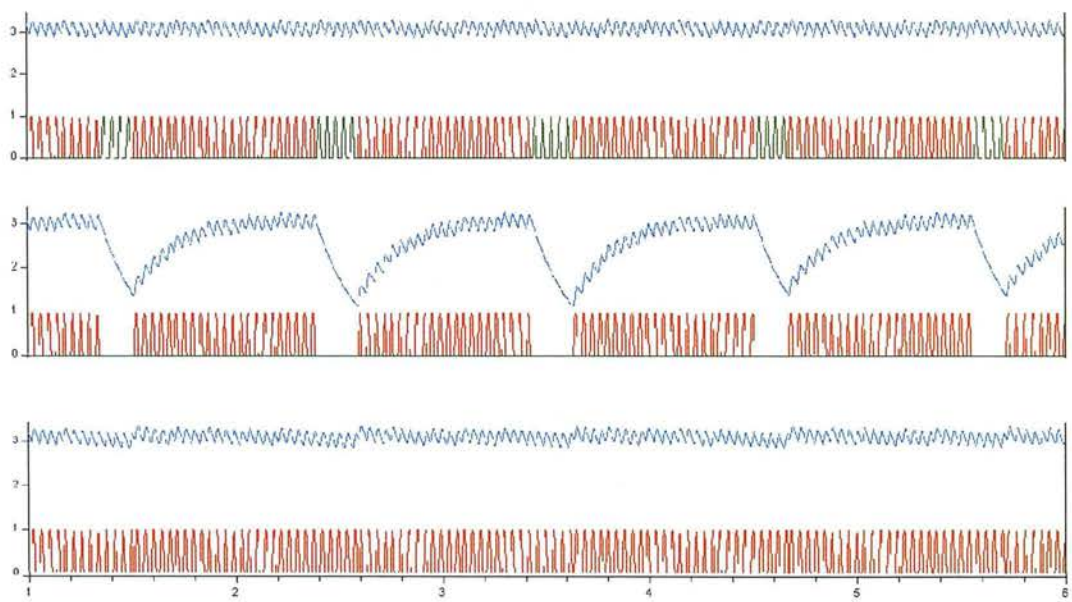
In the course of propel vehicle development, failure of one of the six LPVs occurred occasionally. Because the controller flow algorithm is "open loop", there is no way for the controller to detect such a failure, and so the average flow produced by the pump is reduced in proportion to number of failed LPVs. Unfortunately this degradation is not graceful. A missing LPV introduces an irregularity into the flow of pulses produced by the pump, the significance of which depends on the magnitude of the displacement fraction which is currently commanded.

Consider the case where a 6 cylinder DDP, using the full stroke mode whereby full strokes and idle strokes are mixed, is commanded to deliver 16.66% displacement - exactly one-sixth of full flow. The result will be that a single cylinder will be activated every revolution. At 17.3% of full flow a cyclic pattern is produced that results consisting of each of the LPVs being activated in sequence, with a cycle time of around second. If one of these LPVs does not work, then the result is introduce a very obvious drop-out of flow, of a duration 1/6th of a second, at a period of 1 second.

Results from an experiment are shown below. (Top) the pump pressure trace (bottom) the current drawn by the DDP. The periodic dropouts in the middle section were clearly discernable and quite unpleasant; while there was no apparent difference between the first and the third sections.



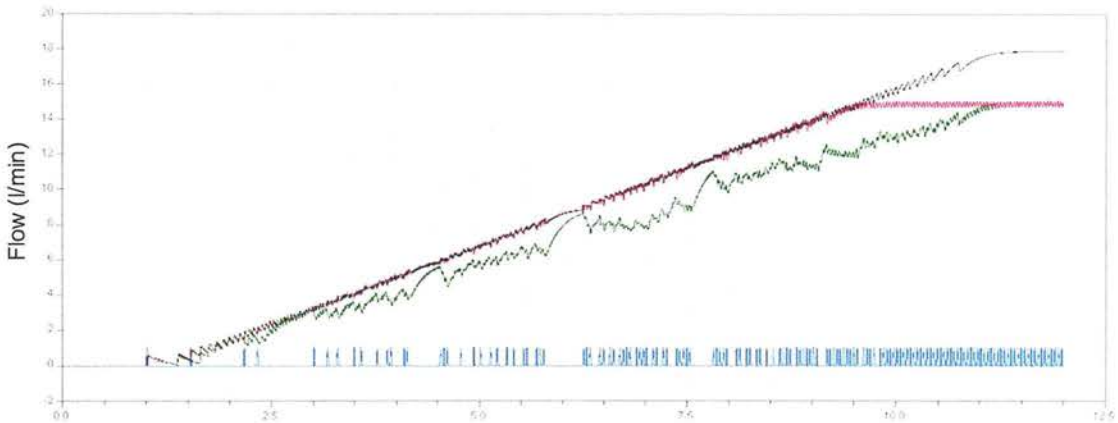
The origins of this behaviour were confirmed using the Dymola simulation. Results below show



Three configurations were investigated - (top) with all cylinders working, (middle) with cylinder #1 faulty, (bottom) with the faulty cylinder #1 disabled in software.

The red line is the sum of DDP flow pulses, while the green line is that portion coming from cylinder#1 .The blue line shows the DDP flow pulses filtered with a first order low-pass of cutoff frequency 0.2s – this gives a similar form to the pressure trace from the experimental result.

This simulation was extended to cover a ramp in demand from zero to full displacement in 10 seconds. Again the flow was filtered with a first-order low-pass (τ=0.02s). Black – normal, green – broken valve, pink- disabled valve, blue –flow coming from cylinder #1 when the pump is working normally, which is missing when #1 is broken. Clearly the dropouts in the green trace occur whenever the broken cylinder is used by the controller. Despite having a gap in its possible pulse sequence, the magnitude of pulsation with the disabled valve is a huge improvement over that of the broken valve.



It is obvious from both simulation and experiment that the controller can vastly improve the behaviour of a DDP with a broken valve if it knows that there is a problem, and therefore there is value in being able to identify broken valves from some feedback from the system. Two methods present themselves:

1. A continuous monitoring of the pressure signal, trying to correlate pulsation in the signal with known patterns of enabling of each valve.

2. A special self-test sequence, perhaps initiated on a regular basis, where the DDP would run with each of the cylinders disabled in turn for a period of seconds. The pulsation could be measured by integrating the magnitude of the high-frequency component of the pressure signal for the period for which the candidate cylinder is disabled. If there were a substantial improvement in pulsation with one of the cylinders disabled, that cylinder would be considered to be suspect and permanently disabled until the machine could be repaired. The above results indicate that such a self-test on a healthy DDP would be transparent to the driver, as long as it was inhibited whenever displacement demand was more that 5/6th.